

# Video tracking of *Drosophila* gravitaxis

Finlay J. Stewart

Neuroinformatics MSc (by research) project

Supervisors: Dr Dean A. Baker & Dr J. Douglas Armstrong

Neuroinformatics Doctoral Training Centre

Institute of Adaptive and Neural Computation, School of Informatics

University of Edinburgh

August 2005



## **Abstract**

Gravitaxis, the movement of an animal in the direction of gravity, has been studied in *Drosophila melanogaster* since the 1960s. It has traditionally been measured using a 'gravity maze' paradigm, where flies choose to walk up or down at a series of junctions and are scored for gravitaxis according to the height at which they exit the maze. However, little is known about behaviour inside the maze. This study addresses that problem by developing a computer vision system for the tracking of multiple flies in the maze, and software that automatically analyses these traces to detect various behavioural events, e.g. junction decisions, falls and U-turns. The behaviour of ten strains is characterised using this system, including wild-types, artificially selected positive and negative gravitactic lines, mutants identified as having a specific gravitactic phenotype, and the learning-defective mutant *rutabaga*. Various differences between strains are found. A spatial analysis reveals that behaviour in many strains differs between the first and second halves of the maze; however the observation that learning mutants do not exhibit this pattern leads us to tentatively suggest that it may be caused by reduced motivation due to decreasing novelty. Markov models of the strains are constructed and used to run a Monte Carlo simulation of gravity maze performance. The results approximately but imperfectly match observed data, suggesting that our formalisation may be missing some important details. Possible improvements and extensions to the software, as well as suggestions for future research based on our preliminary observations, are discussed.



# Contents

<b>1. Introduction and background</b>	
1.1: Introduction	7
1.2: Previous work	
1.2.1: <i>Drosophila</i> gravitaxis	7
1.2.2: Automated animal tracking	9
1.2.3: Commercial video tracking packages	10
1.3: Motivation and aims	11
<b>2. Biological methods</b>	
2.1: <i>Drosophila</i>	13
2.2: Gravity maze	14
2.3: Control experiments	15
<b>3. Video tracking</b>	
3.1: Introduction	19
3.2: Hardware	19
3.3: Software	
3.3.1: Acquiring background image	19
3.3.2: Identifying junctions	20
3.3.3: Detecting potential flies	21
3.3.4: Matching blobs to traces	22
<b>4. Automated trace annotation</b>	
4.1: Introduction	25
4.2: Implementation	
4.2.1: Overview	25
4.2.2: Junction events	27
4.2.3: Stops	28
4.2.4: Falls	28
4.2.5: U-turns	28
4.2.6: Post-processing	28
<b>5. Results</b>	
5.1: Event frequencies	
5.1.1: Method	29
5.1.2: Discussion	29
5.2: Correlations of events with exist scores	33
5.3: Spatial analysis	
5.3.1: Method	34
5.3.2: Discussion	34
<b>6. Modelling behaviour</b>	
6.1: Introduction	37
6.2: Simulating the maze	37
6.3: Constructing the model	38

6.4: Results and discussion	39
<b>7. Discussion</b>	
7.1: Conclusions	43
7.2: Evaluation	43
7.3: Future work	
7.3.1: Improving visualisation facilities	44
7.3.2: Including time information	45
7.3.3: Investigating interactions	45
7.3.4: Investigating the role of novelty in motivation	46
7.4: General discussion	46
<b>Appendix: Pseudocode</b>	
A.1: Video tracker	49
A.2: Trace annotator	51
<b>References</b>	55

# 1. Introduction and background

## 1.1: Introduction

The ability to perceive the direction of the gravity vector is of considerable importance to the vast majority of animal species, not least those that fly. Although we have some insight into the sensory organs involved in gravity perception in organisms as diverse as jellyfish and humans (see Anken & Rahmann (2001) for a review), very little is known about the genetic or neural mechanisms underlying gravity-related behaviour..

Owing to their ease of handling, rapid reproductive cycle, various useful genetic properties (see Beckingham *et al*, 2005) and the wealth of knowledge and tools accumulated over decades of research, the fruit fly *Drosophila melanogaster* is a popular model organism in a wide variety of fields of study. Gravity perception is no exception. In fact, seminal studies by Jerry Hirsch and collaborators in the early 1960s demonstrated a genetic basis for gravitaxis (i.e. attraction towards the direction of gravity) in *Drosophila* – showing for the first time that a high-level behaviour could be influenced by genetics. To quantify gravitaxis they developed the 'gravity maze', a vertically oriented network of tubes with multiple exits at different heights. It remains the dominant assay for gravitaxis to this day..

This project seeks to improve our understanding of *Drosophila* gravitaxis by shedding light on what behaviours are actually taking place inside this maze. This is achieved by recording video footage of flies in the maze, then using software we have developed to automatically detect their positions and analyse their movements over time. By comparing strains of various types, we attempt to elucidate the ways in which various behaviours contribute to the gravitaxis score, which in turn might help us to make links with what is known about the genetic processes in question.

In chapter 2, we describe the experimental methodology, including the strains investigated and control experiments performed to ensure that behaviour in the maze modified for video tracking was similar to that observed in the traditional paradigm. Chapters 3 and 4 describe the implementational details of the tracking and behaviour analysis programs respectively, and also describe the processes of their development. Our results are presented in chapter 5, and chapter 6 describes our attempt to probabilistically model and simulate the various strains' behaviour. Finally, in chapter 7 we discuss the achievements of the project and suggest directions for future work. For the remainder of this section, however, we shall review previous work, both in *Drosophila* gravitaxis and in the automated tracking of animals. We will then discuss the motivations behind the current project and set out the aims of the study.

## 1.2: Previous work

### 1.2.1: *Drosophila* gravitaxis

Gravitaxis, or *geotaxis* as it was until recently more commonly (and somewhat inappropriately) known, refers to the movement of an animal in the direction of gravity (i.e. downwards). Wild-type *Drosophila* have an anti-gravitactic instinct, as evidenced by the well-known observation that they will climb the walls of a vial after being knocked down. To provide a means of quantifying the polarity and magnitude of this gravitactic tendency, Hirsch (1959) developed the gravity maze. This is a maze with a single entrance and many exits, consisting of multiple junctions at which flies must choose to go either up or down, whilst being enticed through the maze by attraction to a light source positioned near the exits

(see section 2.2). The lower the exit from which the fly eventually emerges, the stronger its gravitactic instinct it is deemed to be.

Hirsch and collaborators realised that wild-type populations exhibit considerable variability in gravitactic response. They selectively bred flies which achieved both the highest and lowest gravitaxis scores. Within 48 generations, they had produced two populations with scores substantially higher and lower than the original unselected population (Hirsch & Erlenmeyer-Kimling, 1961). Thus they had shown that the behaviour had a genetic basis, but due to the immaturity of the field of genetics at the time, they were unable to pinpoint any specific genes. They did, however, attempt to discover loci at the level of the chromosome. They achieved this through the use of flies with 'balancer chromosomes' – ones containing multiple inversions rendering them unable to recombine with a homologous chromosome partner (see Beckingham *et al*, 2005). Thus they could breed flies with either one or two copies of each chromosome taken from a known population, and therefore quantify the effects on gravitaxis of each of the three major chromosomes (X, 2 and 3; *Drosophila* also have a tiny 4th chromosome). Perhaps unsurprisingly (with the gift of hindsight), they found that all three contributed to the behaviour, and that all three had responded to selection for gravitaxis, i.e. each chromosome taken from the low selected population individually reduced gravitaxis scores compared to its wild-type counterpart (Erlenmeyer-Kimling & Hirsch, 1961). This led them to conclude that “there are genes distributed over most of the genome which influence the response to gravity” (p1069).

Dobzhansky *et al* (1974) used the same gravity maze paradigm to breed extreme gravitactic populations in the related species *Drosophila pseudoobscura*, and showed the effects were robust to changes in temperature, age of fly and light intensity, although interestingly qualitative differences were observed when red light was used.

Hirsch *et al* continued to intermittently apply artificial selection to their high and low lines into the 1980s. They found that the populations stabilised, i.e. would retain their gravitactic behaviour even when selection was relaxed (Ricker & Hirsch, 1985). In fact, the low line (and perhaps the high one as well, though the data is less conclusive) returns to its extreme gravitactic behaviour after a period of reverse selection (Ricker & Hirsch, 1988). These findings suggest that the stability is not merely due to a decrease in genetic variation due to inbreeding, but rather that a new coadaptation of genes has evolved such that genetic homeostasis will resist any drift back towards the wild-type gravitational phenotype.

Ricker & Hirsch (1988) conducted a statistical analysis of gravitaxis scores after breeding between selected and unselected populations, and concluded that only one or two loci (each consisting of either a single gene or a group of closely linked genes) contributing to gravitaxis existed on each chromosome. Stoltenberg *et al* (1995) surveyed 22 allelic markers in the *Drosophila* genome, and found that three reliably differed between the high and low selected lines. This could be due to either a gene at or near the marker being responsible for the phenotypic effect, or random drift. Differentiating between these possibilities was addressed by Stoltenberg & Hirsch (1996). They interbred selected and wild-type flies for 66 generations, as this would cause associations between markers and gravitaxis due to genetic drift alone to disappear. They found that just one marker persisted in the most gravitactic flies: the *Adh* gene on chromosome 2. Thus a gene (or possibly genes) in the close vicinity of *Adh* affects gravitaxis.

The dawning of the genomic era meant that genes related to gravitaxis (or any other phenotype) could

be searched for much more effectively. A key study was that of Toma *et al* (2002), who used microarrays to measure the differences in mRNA expression in the heads of high and low selected lines. This yielded around 250 candidate genes, but many of these are likely to be due either to drift or 'hitch-hiking' by proximity to a genuinely selected gene. They therefore tested gravitaxis in either null or extreme hypomorph mutants for a selection of the identified genes. In this way, they showed that the genes *cry*, *Pdf* and *Pen* have a direct effect on gravitaxis.

Armstrong *et al* (2005) take a different, and complimentary, approach to identifying genes involved in gravitaxis. They screened around 300 different mutant strains for gravitaxis, and identified those that exhibited significantly higher or lower scores than wild-types. Twenty-three of these were found to have a specific gravity phenotype, in that they scored normally on other behavioural tests such as courtship and locomotion. Thus 18 genes were implicated with gravitaxis. This high number of genes may initially seem at odds with Ricker & Hirsch's (1988) estimates. However, those were based on just two selected populations; it is very likely that many other mutations could affect gravitaxis but never occurred in the ~600 generations of evolution these populations underwent. Interestingly, one of the mutations identified by Armstrong *et al* (2005) affects a gene very close to *Adh*, which they named *yuri*. Transgenic rescue confirmed that decreased levels of *yuri* cause decreased gravitaxis (i.e. more climbing).

We are therefore at the stage of having identified several genes linked to gravitaxis. The mechanisms by which these genes affect such a high-level behavioural response remain mysterious. Many of them have unknown functions, but the better understood ones to give some tantalising clues. For example, *off-track* - identified in Armstrong *et al*'s (2005) screen and further investigated in this study – is involved in neural pathfinding and cell adhesion. Looking at anatomical expression patterns can also yield some interesting observations. *Yuri*, for instance, is expressed in a subset of mechanosensory neurons in the second antennal segment and their axons projecting into the CNS.

### **1.2.2: Automated animal tracking**

Traditionally, animal behaviour has been measured by means of an experimenter manually recording positions, actions, etc., usually from video footage. This process is tedious and typically inaccurate, due to observer fatigue and/or drift, i.e. the tendency of observers to alter their criteria for recognising a particular behaviour over time. As a result of the somewhat subjective nature of behavioural classification, inter-rater reliability can be rather poor (Spruijt *et al*, 1998).

To overcome these problems, various attempts have been made to automate the analysis of behaviour. Diverse approaches to tracking such as grids of infrared beams or Doppler radar (see Noldus *et al*, 2002) have been employed, but the majority of work has focussed on video tracking. The method is flexible and cheap (as it requires no specialist equipment), and increasingly simple thanks to the emergence of commercially available tracking software (see below).

The most basic trackers (e.g. Fourcassie & Traniello, 1995) work by thresholding each frame of video based on pixel brightness (or colour). As a white background is typically used, the largest region of connected subthreshold pixels is assumed to be the animal in question. The centroid of this region is recorded at each frame to give a trace of positions over time. Such systems are restrictive, in that they can only track one animal at a time, and the environment must be highly controlled and artificial. Tracking multiple animals simultaneously is useful as it increases experimental throughput, and obviously essential if one wishes to study interactions between individuals.

Voss & Zeil (1995) implemented an impressively sophisticated system (especially considering the available technology) that could track multiple flying wasps in an outdoor environment. Regions potentially containing wasps were found by detecting differences between consecutive frames, i.e. moving objects. These pixels were then grouped into 'blobs' using a scanning algorithm that compared results obtained at various difference thresholds. Having found these blobs, they then had to solve the *correspondence problem*, i.e. determining how to match these to the insects detected in previous frames. They solve this problem by using physical constraints such as maximum velocity, acceleration and turning angle to obtain the most probable assignment. Finally, they employ a pattern matching algorithm to estimate the body-axis orientation of each wasp.

A system for tracking multiple ants is presented by Balch *et al* (2001). They employ the technique of *adaptive background subtraction* to detect potential insects. This is similar to frame subtraction (see above), but uses the average of many previous frames and thus tends to be more robust. Their approach to the correspondence problem was to use a simple minimum distance matching algorithm between consecutive frames. The system was rather poor at handling occlusion, i.e. two or more ants overlapping. They address this problem in later work (Khan *et al*, 2004) by developing a highly sophisticated particle filter system based on probabilistic models of ant movement. By incorporating this prior knowledge they are able to infer insect movements when occlusion prevents them from being tracked directly.

Heward (2005) developed a system for automatically analysing *Drosophila* courtship behaviour. Naturally, dealing with occlusion is of utmost importance in this situation. He took the approach of monitoring, based on previous frames, which blobs could potentially contain more than one fly. When an occlusion event was detected, a K-means clustering algorithm was employed to split the appropriate blob(s) into its component flies. Courtship behaviour was detected by a rule-based classifier, trained on manual annotations by experts, using parameters such as the relative distances and angles between flies.

An example of a multi-target tracker operating in rather challenging conditions is that of Walther *et al* (2004). They describe a system for tracking marine animals from a remotely operated underwater vehicle. The task is complicated due to the translucent nature of the gelatinous animals they attempt to detect, the fact that the vehicle is moving, the highly variable lighting conditions and the abundance of noise in the form of floating organic debris. They too use adaptive background subtraction, but with a fairly short timescale as the scene is not static. Potential animals are detected by the use of several *feature maps* which identify regions of brightness and colour contrast and organised spatial orientation. Combining these maps yields a single *saliency map* giving positions of interesting objects. These objects are then tracked using a Kalman filter taking into account the vehicle's movement.

### **1.2.3: Commercial video tracking packages**

A number of software packages for tracking animals in behavioural experiments are now available. The leading product is Noldus' *EthoVision* ([www.noldus.com](http://www.noldus.com); Noldus *et al*, 2002), which claims to offer a general purpose solution suitable for any animal or paradigm. The most advanced version allows tracking of multiple separate arenas each containing up to 16 animals. It offers greyscale, colour and background subtraction based detection, and allows the user to specify zones of interest of which animals' entering and leaving is recorded. Facilities for visualization and analysis are also included. Several studies have made use of the software: e.g. Belmain *et al* (2000), Martin (2004).

Other commercial programs include:

- Biobserve's *Viewer* ([www.biobserve.com](http://www.biobserve.com)), which offers similar functionality to *EthoVision* but is more geared towards larger animals such as rats.
- Qubit Systems' video tracking software ([www.qubitsystems.com](http://www.qubitsystems.com)), which only allows a single animal to be tracked.
- Solltech's *DIAS* (Dynamic Image Analysis System; [www.geocities.com/solltech/dias](http://www.geocities.com/solltech/dias)) which is primarily designed for analysing the morphology of groups of cells, but has been adapted for use with multiple *Drosophila* (Wolf *et al*, 2002).

We have chosen to develop our own software rather than use a commercial package. This decision is motivated partly by cost, but also because it was thought that such a program would not meet our requirements. Despite being the most sophisticated package available, *EthoVision* is reportedly poor at tracking small insects, especially at the range we are to use, where a fly may be only ~3 pixels wide. It also has difficulty tracking multiple interacting animals due to occlusion (Heward, 2005).

### 1.3: Motivation and aims

The vast majority of work on *Drosophila* gravitaxis has made use of Hirsch's gravity maze assay (although more crude measurements, such as speed of climbing, have also been employed (e.g. Strauss & Heisenberg, 1993)). A criticism of this methodology is that it is something of a 'black box' – we only record how high up flies exit the maze, with no insight into what they are doing while inside it. The possibility that factors other than gravitaxis are influencing exit score is therefore a key concern.

Efforts have been taken to rule out trivial explanations for supposed gravitactic behaviour. In Armstrong *et al*'s (2005) genetic screen, the mutants were tested to ensure that their courtship and flight response were within normal limits. Flies with obvious anatomical defects were excluded, as were any strains where >20% failed to complete the gravity maze in 3 hours. Exit scores in the maze were compared to those obtained when the same maze was oriented horizontally to demonstrate that gravity was affecting their behaviour. Finally, all flies exhibited the reflex of climbing the walls of a vial after being knocked down, demonstrating that they had the ability to detect gravity and climb against it, at least in a stress situation.

Even with all these controls, some doubts remain about what the gravity maze is actually measuring. For instance, a low score could be caused by flies frequently falling, or by them 'choosing' to walk down at junctions. These two phenotypes could have very different neural underpinnings, yet the gravity maze would be unable to differentiate between them. Even among flies that fall frequently, a low score could represent either a lack of perseverance or a particularly strong phototactic instinct causing the flies to progress through the maze rather than attempting to recover following a fall. The objective of this project is therefore to look inside the maze and characterise the types of behaviours exhibited by various strains of fly. It is hoped that this will yield some clues as to the factors that can influence exit scores, which might in turn help us to understand gravitactic behaviour in light of what we know about the various genes identified.

In fact, some work of this kind has already been carried out. In a preliminary unpublished study, flies were tracked in the first half of the gravity maze. Their choices at junctions (along with some other parameters) were recorded, allowing a probabilistic model to be created. However, it was found that when this was used to simulate behaviour in the full maze, the exit scores did not match the data. Thus it was hypothesised that some kind of learning was occurring as flies negotiated the maze, with the

result that their behaviour in the latter half differed from that in the former. Specifically, it was hypothesised that every fall a fly experienced made it less likely to attempt to climb back up following the fall (J.D. Armstrong, personal communication).

The detailed aims of the current project are to:

- develop apparatus and software that will allow flies to be automatically tracked whilst completing a gravity maze.
- demonstrate that flies' behaviour in this experimental paradigm is comparable to that of previous gravitaxis assays.
- develop software that will analyse traces obtained of flies' movements to produce high-level behavioural descriptions.
- characterise and compare the behaviour of several different strains, including wild-types, high/low selected lines, gravity mutants identified in Armstrong *et al*'s (2005) screen, and learning/memory mutants.
- test the hypotheses suggested by the previous pilot study (hence the inclusion of memory mutants).
- model and simulate behaviour in the gravity maze, and compare this real data to evaluate the degree to which we have captured the factors affecting gravitaxis.

## 2. Biological methods

### 2.1: *Drosophila*

Fruit flies of the species *Drosophila melanogaster* were used in all experiments. All *Drosophila* were raised on standard corn meal medium at 25°C with a 12hr light/12hr dark cycle. Young males (aged <24hr) were collected under CO<sub>2</sub> anaesthesia and aged in vials for 48hrs at 18°C before being used in experiments. Individuals with obvious anatomical defects were discarded. All experiments were carried out during the day using naïve flies.

We shall now describe the various strains used in the study:

#### **Canton-S (*cs*) and Oregon-R (*or*)**

These are two wild-type strains. It is useful to investigate more than one, as behavioural responses can vary markedly between different wild-types. For instance, *cs* flies will consistently jump in response to a small amount of odourant, whereas Samarkand (another wild-type strain) seldom do (Anholt & Mackay, 2004). Even *cs* stocks from different labs are found to differ in their gravity maze scores (see section 2.2) (Armstrong *et al*, 2005).

#### **Hi5 (*jhh5*) and Lo (*jhl*)**

These are the two lines produced by J. Hirsch and collaborators (hence the 'jh' prefix) by selectively breeding flies with strong anti-gravitactic (upward movement) and gravitactic behaviour, respectively (see section 1.2.1). They have stabilised, meaning that the gravity phenotype persists indefinitely even in the absence of selection.

#### ***rut2080***

The gene *rutabaga* (*rut*) encodes a protein that catalyses the production of cyclic AMP (cAMP) from ATP. In the adult fly, is expressed primarily in the mushroom bodies of the brain (Han *et al*, 1992). It has a well-established role in learning and memory, as *rut* mutants perform poorly at avoiding an odour previously associated with an aversive stimulus (Aceves-Pina & Quinn, 1979), as well as at tests of reward learning (Tempel *et al*, 1983) and habituation and sensitization (Duerr & Quinn, 1982). The line used in this study is homozygous for the mutant allele *rut2080*, which has a learning/memory phenotype. It is included in this study to investigate the possibility that a fly's memory of experiences early on in the gravity maze could affect its gravitactic behaviour later on.

#### **82y, 144y, 179y, 248y and c255**

These strains were all identified as having a specific gravity phenotype in Armstrong *et al*'s (2005) forward genetic screen (though *c255* was not reported). We shall briefly describe what is known about the gene disrupted by the P{GawB} insertion in each case. 'Up' and 'down' denote negative and positive gravitaxis, respectively.

82y (up): 108bp upstream of start of *off-track*, a receptor tyrosine kinase involved in neural pathfinding and cell adhesion.

144y (down): 132bp upstream of start of *CG6330*, a putative uridine/purine phosphorylase.

179y (up): 572bp into first exon (5' UTR) of *CG11940*, a putative PH- and RA- domain-containing homologue of Grb molecular adaptors.

248y (up): ~1kb into 4kb first intron of *discs large 1*, a guanylate kinase containing PDZ, SH3 and P-loop domains, with roles in septate junction formation and synapse structure.

C255 (down) in an intron of *Gap1*, a Ras GTPase-activating protein.

Additionally, some experiments were performed using a *CyO* line with considerable locomotor defects, the intention being to characterise the differences in behaviour between flies with low exit scores due to gravitaxis v.s. those simply unable to climb properly. However, work on this strain was abandoned as the majority of flies failed to exit the maze in the allotted time.

## 2.2 Gravity maze

Mazes of the type previously described by McMillan & McGuire (1992) were used as a gravitaxis assay. The maze measures approximately 35cm by 35cm. It has a single entrance and nine exits (see fig 2.2.1), which the fly reaches by choosing to go up or down at a series of eight T-junctions (assuming no backtracking). The maze is made from sections of 5mm transparent plastic tubing joined by translucent white T- and Y-shaped connectors and mounted on a rigid board. The free arms of the Y-connectors along the top and bottom edges of the maze are blocked off with cotton wool. Note that the narrowness of the tubing makes flight impossible, so the flies must walk. Flies finishing the maze are collected in test tubes attached to the exit tubes via a glass pipette tip, preventing them from leaving the test tube once inside. A fluorescent light is positioned vertically at the exits to encourage the flies to progress through the maze by phototaxis.



*Figure 2.2.1: The gravity maze, viewed from the recording camera (though the video footage is not in colour). Flies enter at the the left and progress to the test tubes on the right, attracted by a light gradient (which can be seen in the picture). Exits are numbered 1 (bottom) through 9 (top). Note that the T-junctions appear in 8 columns, which we shall refer to as column 1 (left, consisting of just one junction) through column 8 (right).*

Previous work has used a number of these mazes arranged side-by-side in a temperature-controlled cabinet. In order to allow for video tracking, some modifications were made to this paradigm. A single maze was set up in a darkened, non-temperature-controlled room. To make tracking possible, it was necessary to increase the ambient light by the addition of a small secondary fluorescent light oriented horizontally and positioned ~50cm above the camera. Additionally, the maze was put on a white background with a number of black dots to facilitate automatic recognition of the junction points (see section 3.3.2). Control experiments were carried out to ensure that the flies' behaviour (i.e. exit scores) were similar despite these alterations (see section 2.3).

When performing experiments, a number of males (6 in tracking experiments, up to 26 in controls) would be transferred to a piece of tubing ~5cm long and 6mm in diameter, blocked at both ends with cotton wool. Approx 5min later (to allow the flies to recover and become accustomed to the new environment) this tube would be connected to the entrance of the maze. The connection was initially made via a plastic connector similar to those used for the junctions. However, this was changed to a narrow glass tube ~3cm long, as this allows only one fly to enter the maze at a time, improving tracking accuracy (see section 3.3.4). This apparatus has the added advantage of making it easier to quickly attach the tube containing the flies to the maze.

Only males were used, primarily to prevent courtship from occurring in the maze which would obviously confound the data. Females could have been tested separately, but they may have left olfactory cues which would influence the behaviour of later males in the maze. Control experiments have demonstrated that in the male-only paradigm employed here, no effects on exit score due to olfactory cues are found (J. D. Armstrong, personal communication).

### 2.3 Control experiments

As discussed above, it was impossible to carry out video tracking experiments in exactly the same conditions as previous gravity assays. In order for findings of this study to be relevant to previous work, it was important to ensure that the flies behaved similarly across the two environments. Control experiments were therefore undertaken to compare the exit profiles of various strains across the two conditions. We assume that similar exit profiles indicate that the flies are behaving similarly in the maze, although of course we have no way of measuring detailed behaviour in the old paradigm.

The strains *cs*, *rut*, *jhh5* and *jhl* were used for the comparison. Initially, the video tracking maze was located in an ordinary lab environment, with just one fluorescent light placed at the exits. Between 12 and 26 flies were put through the maze at a time, either in the 25°C cabinet or the video tracking maze. The latter was set up exactly as it would be for recording, though no footage was taken. Each trial lasted ~45min; if >20% of flies failed to finish in this time the trial was discarded, but this happened infrequently.

These experiments were abandoned after 3-4 video tracking trials for each strain, as it quickly became apparent that the exit scores were dissimilar across the experimental settings; the video tracking maze

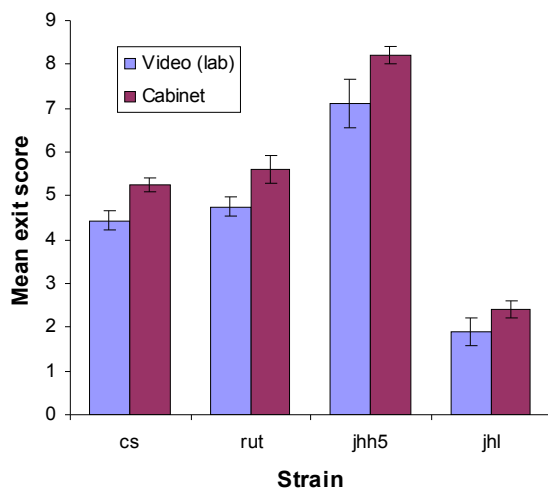


Figure 2.3.1: Mean exit scores, bars represent  $\pm 1SE$ , calculated treating each trial as an independent measurement. For the video tracking data,  $n_{trials} = 3-4$ ,  $n_{flies} = 64-78$  for each strain.

seemed to pull scores downwards (see fig 2.3.1). Exactly why this should be the case is unclear, but we hypothesised that the high level of ambient light was interfering with the flies' phototactic attraction towards the exits and thus altering their scores.

The maze was therefore moved to a dark cupboard and set up as described above, and the procedure was repeated. Not only was the level of ambient light lower, it was now constant due to the room having no windows, unlike the lab. Between 8 and 10 trials were performed for each condition (strain x environment); 126-241 flies completed the maze in each. Exit profiles are given in fig 2.3.2.

Figure 2.3.3 compares the mean exit scores for each strain. In assessing whether differences are significant, we can treat each fly as an independent measurement, giving us a very high df. However, this is perhaps inappropriate as several flies are simultaneously in the maze, and could potentially interfere with each other. We can therefore treat the mean exit score from each trial as an observation, giving us a cleaner test but less statistical power. Using the former method, only *jhh5* exhibits a significant difference between the two environments (*jhh5*:  $t(395)=2.696$ ,  $p<.01$ ; *cs*:  $t(344)=1.434$ , ns; *jhl*:  $t(378)=1.142$ , ns; *rut*:  $t(380)=0.706$ , ns). With the latter, none of the differences are significant, but this could potentially represent type II errors occurring due to a poverty of data (*cs*:  $t(16)=0.964$ , ns; *jhh5*:  $t(17)=1.535$ , ns; *jhl*:  $t(16)=0.628$ , ns; *rut*:  $t(16)=0.706$ , ns).

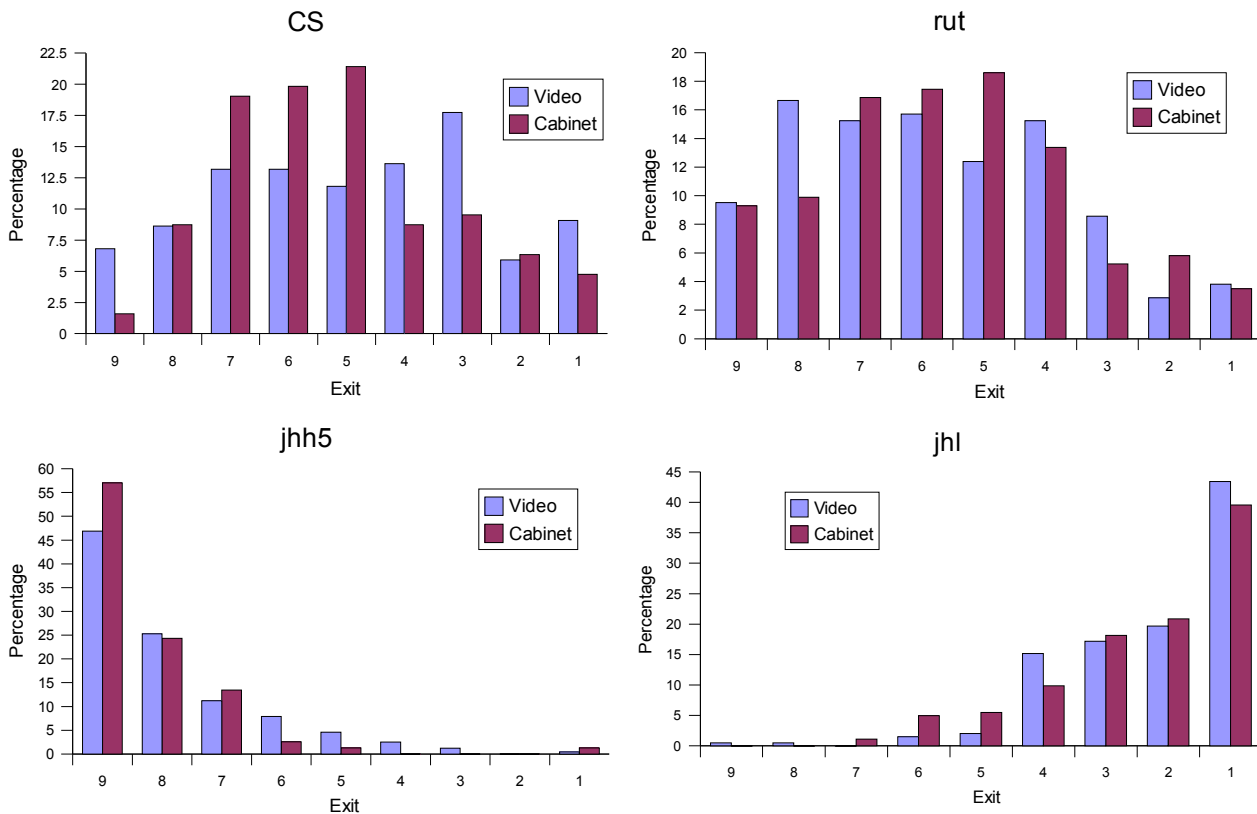


Figure 2.3.2: Exit profiles for the cabinet and video tracking set-ups. Note that they are similar between the two environments, although there is perhaps a tendency for higher variability in the video tracking assay. *cs*, video:  $n_{trials}=10$ ,  $n_{flies}=220$ ; cabinet:  $n_{trials}=8$ ,  $n_{flies}=126$ . *rut*, video:  $n_{trials}=10$ ,  $n_{flies}=210$ ; cabinet:  $n_{trials}=8$ ,  $n_{flies}=172$ . *jhh5*, video:  $n_{trials}=10$ ,  $n_{flies}=241$ ; cabinet:  $n_{trials}=9$ ,  $n_{flies}=156$ . *jhl*, video:  $n_{trials}=9$ ,  $n_{flies}=198$ ; cabinet:  $n_{trials}=9$ ,  $n_{flies}=182$ . Note that  $n_{flies}$  refers to the number of flies completing the maze; those that do not are ignored.

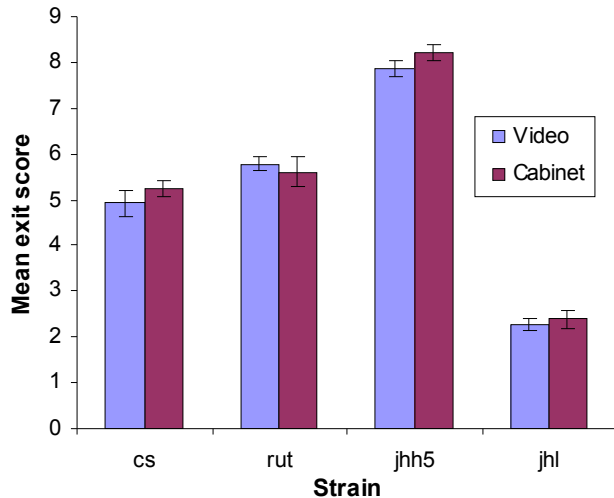


Figure 2.3.3: Mean exit scores. Bars represent  $\pm 1SE$ , calculated treating trials (i.e. not flies) as independent measurements.

For strains *cs*, *jhl* and *rut* we can be reasonably confident that the differences between the traditional cabinet set-up and the new video tracking one are not affecting the flies' behaviour, at least in terms of exit scores. The finding that there is a difference for *jhh5* is a little troubling. However, as the magnitude of the difference is fairly small (0.35 exit units) it seems reasonable to assume that their behaviour is not radically altered.

These controls have only used a subset of the strains that we shall go on to analyse. We cannot rule out the possibility that other strains will be affected differently. However, the selected strains do represent a varied sample covering wild types, memory mutants and extreme gravitactic and anti-gravitactic strains.



## 3. Video tracking

### 3.1: Introduction

We developed video tracking software which would take as input a video of several flies in the gravity maze, detect their positions frame by frame, and track the movement of each individual from one frame to the next. Its output is a series of (x,y) coordinates for each fly, listing its position in each frame of video. We shall now describe in detail how this was achieved, discussing the reasoning behind the various implementation decisions made.

### 3.2: Hardware

An Axis 206M network camera ([www.axis.com](http://www.axis.com)) was used to record video footage in greyscale motion JPEG format at 8 frames/second with a resolution of 1280x1024 pixels. The compression was set to 'medium', resulting in a file size of ~32MB/min. Pilot experiments were carried out using an Axis 205 camera, which has a maximum resolution of 640x480, but this was found to be insufficient to reliably detect flies (which measure ~3mm in length) while monitoring the whole maze.

The footage was recorded on a laptop computer then transferred to a 3GHz PC with 1GB RAM running Windows XP for processing. This was done offline, as the tracker could run at a maximum of ~45% real-time on this system (or slower if the tracked video was displayed on-screen).

### 3.3: Software

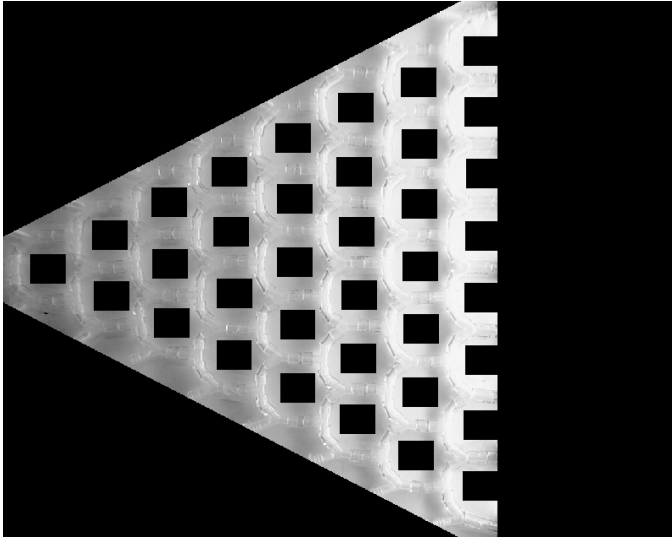
The tracking program was written in Java and is outlined in box 3.3.1. The steps of this algorithm are explained in greater detail below. Pseudocode for the various operations discussed can be found in the appendix (section A.1).

```
Acquire background image
Identify junctions
While flies still in maze:
    Get next frame
    Perform background subtraction to detect 'hit' pixels
    Find biggest 'blobs' of hit pixels
    Match blobs to flies' last known positions
    Record each fly's new position
Save to file
```

*Box 3.3.1: The top-level video tracking program.*

#### 3.3.1: Acquiring background image

The tracking program works on the principle of background subtraction, i.e. each frame is compared to an image of the empty maze and any regions darker than expected are assumed to be caused by flies. In order to make this comparison, the first 5 frames of video (before the flies have had time to leave the entrance chamber) are averaged to produce the 'background' image that will be used throughout the run. This averaging was found to improve accuracy relative to using a single frame, as it helps to minimise the effect of fluctuations in pixel intensity values due to noise. Note that at this and all subsequent stages, parts of the image obviously outside of the maze are not processed (see fig 3.3.1.1). As well as improving efficiency, this increases the program's sensitivity to genuine blobs in the maze (see section 3.3.3).



*Figure 3.3.1.1: Selective processing to improve efficiency and accuracy. The image is a frame of video footage; the blacked out areas are not processed by the tracking program. The three large regions are specified by simple hard-coded boundaries. The small rectangles are added at runtime once the black spots have been located. Each is centred on the underlying spot's CoM, and its dimensions are set according to the universal scale factor (see section 3.3.2).*

### 3.3.2: Identifying junctions

For the later analysis of the flies' behaviour, it is important to record their paths relative to the layout of the maze. Each 'island' in the maze is marked with a black dot for this purpose (see fig 2.2.1). As these dots are darker than their surroundings, some kind of thresholding can be used to detect them. The simplest approach would be to simply set a hard-coded intensity value. However, this was found to be too inflexible. Although conditions were controlled, slight variations in the position of the maze, lights and camera inevitably occurred, which together with the camera's automatic brightness compensation, led to variability in overall brightness and contrast between sessions. A more sophisticated approach was therefore required to calculate a suitable threshold at runtime.

The problem is addressed by calculating a histogram of pixel intensity values (from 0 to 255) in the background image. This yields a small 'dark' peak, corresponding primarily to the dots, and a large 'light' peak, corresponding to the white background (including background seen through the tubes). This is smoothed by replacing each value with the mean of itself and the surrounding 8 values, in order to eliminate minor peaks and troughs. The lowest point of the trough between these peaks is then identified and used as the threshold value (see fig 3.3.2.1a).

An alternative approach considered was that of adaptive thresholding, where instead of the global threshold we have used here, many local thresholds are calculated based on just the surrounding region of the image. The sizeable variation in brightness across the image (due to the light gradient) makes adaptive thresholding an attractive choice, but the simpler global method was found to perform adequately and therefore it was not implemented.

The 36 biggest blobs (see section 3.3.3) of subthreshold pixels are assumed to be the dots (see fig 3.3.2.1b). Measuring the distance between the centres of an adjacent pair of dots gives a measure of scale. The distance between a dot and the T-junction to its left is a constant proportion of this scale factor, as of course is that to the Y-junction to the right. Thus, the positions of all junctions are inferred relative to the dots. The areas around the dots are also excluded from further processing, as they contain no tubes (see figure 3.3.1.1). Note that the Y-connectors along the top and bottom edges of the maze are not considered Y-junctions, as the fly is unable to go up-left or down-left, respectively.

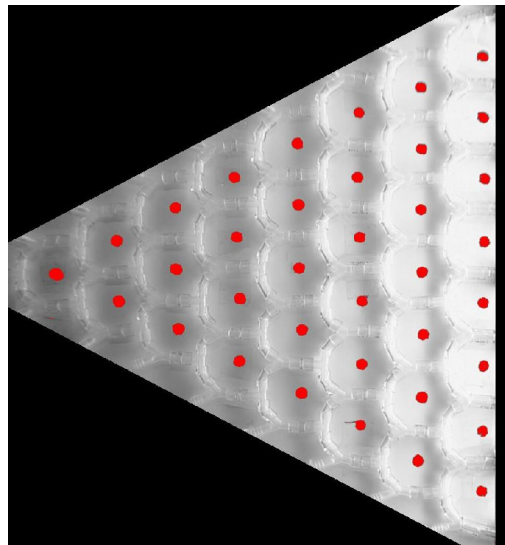
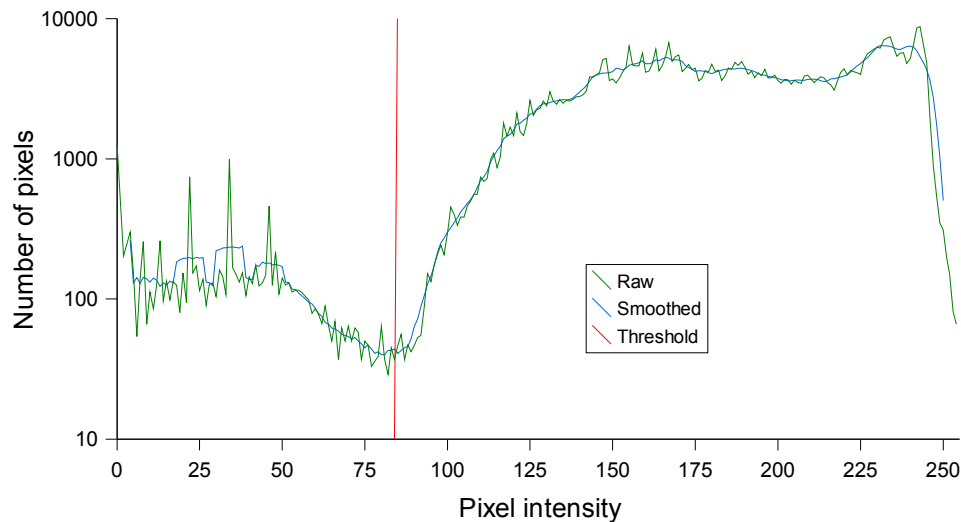


Figure 3.3.2.1: (a, above) A sample pixel intensity histogram, showing the raw and smoothed data and the automatically determined threshold value. Note that the y-axis is logarithmic.

(b, left) The corresponding image, with subthreshold pixels labelled red. One can see that the algorithm has successfully picked out the spots.

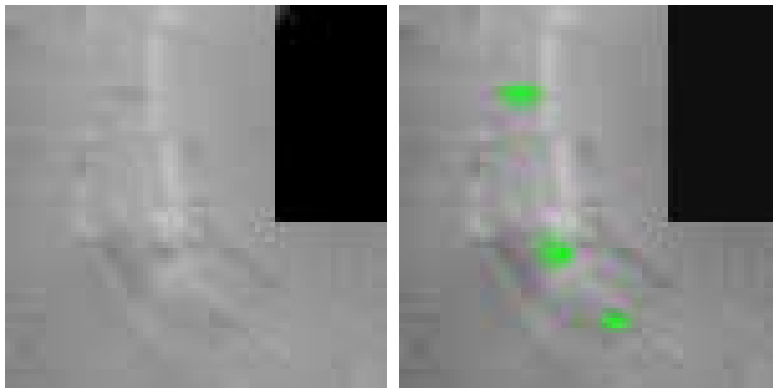
This scale factor is also recorded and used in the annotation program (see section 4.2). This approach is desirable as it avoids hard-coding distance and speed thresholds, making the system robust to small (or even large) variations in the distance between the maze and camera.

### 3.3.3: Detecting potential flies

To detect flies, we make use of the fact that they are darker than the white background. It would be possible to use simple intensity thresholding again, as we did to detect the spots. However, this is a poor approach since a fly viewed through a translucent connector on the right (bright) side of the maze may well be lighter, in absolute terms, than a shadow on the left side, for instance. We therefore use background subtraction, as mentioned above. This method is highly effective in solving problems of uneven lighting and dark objects in the scene, but is very intolerant to camera movement or changes in lighting. A more robust technique is that of adaptive background subtraction, where the background image is continually and gradually updated (see Balch *et al*, 2001). Thus any object that remains stationary for a prolonged period of time (and therefore is probably not an animal) becomes part of the background and is ignored. However, due to the relatively short duration of our experiments (maximum 15 min) and the controlled conditions, this level of sophistication was judged unnecessary.

Each frame of video is compared to the background image, and all pixels which are darker than their background counterpart by at least some threshold value are considered 'hits' (see fig 3.3.3.1). This threshold was set through trial-and-error and represents a trade-off between sensitivity (false negative pixels) and susceptibility to noise (false positives). Thus a boolean map of the image is produced.

This process yields many single pixel hits due to noise. However, if a group of neighbouring pixels are hits, we can be reasonably confident that this represents a fly. A 'blob' is defined as a number of adjacent (up, down, left or right, *not* diagonal) hit pixels. The map is analysed to find all blobs bigger than a threshold size. Five pixels was found to be a suitable lower limit, again representing a tradeoff between tracking noise and missing real flies. In the event that there are more blobs than flies, only the `nflies` biggest blobs are accepted, where `nflies` is the number of flies (typically 6). Additionally, blobs larger than a certain size (42 pixels) are split into two, as it is assumed they represent two partially occluding flies. The centre of mass (CoM) of each blob is calculated, giving at most `nflies` (x,y) positions where flies are thought to be.



*Figure 3.3.3.1: Detecting hits. The left image shows a close up of the original video frame. It contains three flies, although they are very difficult to see. The right image shows all 'hit' pixels marked in green. These are pixels at least 12 intensity units darker than their counterpart in the background (empty maze) image. Three 'blobs' can clearly be seen.*

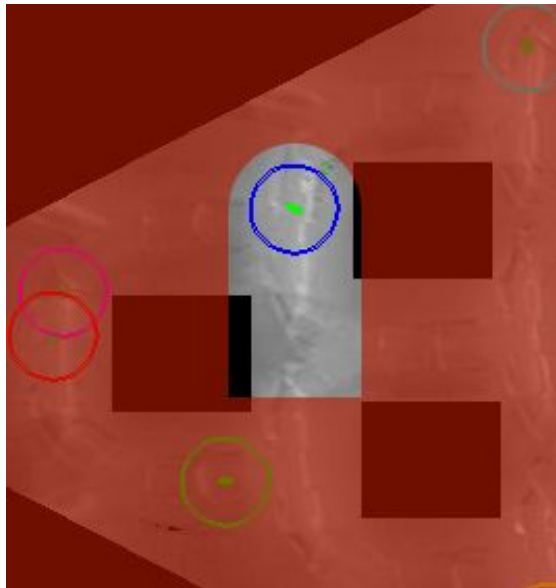
*Erosion* and *dilation* are two manipulations often used to 'clean up' binary images. Erosion means shrinking a blob by removing all pixels on its edge, perhaps for several iterations. This can be useful if two objects are partially occluded, as they will be broken apart at the narrow connecting point. Dilation is the opposite operation, i.e. it grows blobs. This is useful in joining up regions that in fact belong to the same object; for example, a fly and its shadow. Since both kinds of problems occur in the tracking system, it was reasoned that applying either operator would create as many problems as it solved. No image cleanup therefore takes place.

### **3.4: Matching blobs to traces**

Recall that we want to obtain a trace of each *fly's* movements. Thus merely detecting flies is not enough, we must solve the correspondence problem mentioned in section 1.2.2. That is, we must maintain a correspondence between blobs in different frames that represent the same individual in order to build up `nflies` coherent traces. These traces are sequences of the positions of each fly at each frame, with potentially null entries if a fly was lost at any point.

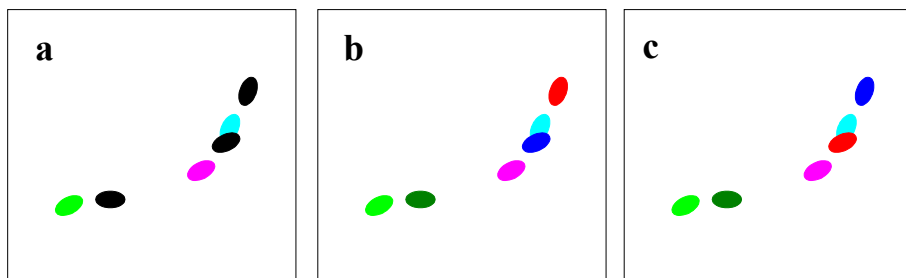
In order to update these traces at each frame, the program must match the blobs to the flies' last recorded positions. This is subject to some constraints: there is a maximum distance at which a match can be made, to prevent traces from jumping between flies (or indeed to spurious blobs arising from noise). Yet again, this parameter represents a trade-off. A tighter matching distance reduces the chance

of traces jumping around, but means that a fly can more easily become permanently lost if it fails to be detected for a few frames. To allow for flies falling, the matching limit is more relaxed if the direction of movement is roughly downwards (see fig 3.3.4.1). The traces are initialised at  $(-5, \text{imageHeight}/2)$  where  $\text{imageHeight}$  is the y-resolution of the image, i.e. 1024. Thus flies entering the image can be successfully picked up.



*Figure 3.3.4.1: Matching blobs to traces. The image is a close-up of a video frame. Hits are once again marked in green. Coloured circles are the program's representation of the current positions of the traces. The trace represented in blue is successfully tracking a fly. The area to which this trace would be allowed to move at the next frame has been marked (i.e. the rest of the image is shaded maroon). These generous bounds allow for the fact that a fly may occasionally go undetected for several frames.*

Initially, we matched blobs to traces using a greedy algorithm, i.e. the closest pair were matched first and removed from the search, then the next closest pair, and so on. However, this was found to be unsatisfactory, as traces would occasionally switch flies in situations like the one illustrated in fig 3.3.4.2. We therefore use a depth-first search to find the globally best assignment. This is defined as the one that minimises the sum of Euclidean distances between traces and matched blobs, with an arbitrary large penalty for each null match.



*Figure 3.3.4.2: Greedy vs. global matching. In this toy example, (a) shows three flies' current positions (black) and the positions of the traces at the last frame (colours). (b) shows the results of greedy matching. The light blue trace would match first, as it has the closest blob in the new frame, then the green, and finally the red, which one can see is a poor match. (c) is the optimal assignment, which would be found by depth-first search.*

Conducting an exhaustive search is rather computationally expensive – in fact its naïve implementation reduced the tracker's speed to  $<10\%$  real-time. An extra constraint was therefore added: a trace may only match with the one of its 3 closest blobs (or null). This heuristic markedly reduces the size of the search tree and thus improves performance, although it sacrifices guaranteed optimality as the search is

no longer exhaustive. However, in the vast majority of situations the search space will include the optimal assignment. As a further performance-improving measure, partial assignments are aborted as soon as their cost exceeds that of the current best assignment.

Ideally, each completed trace should track the same fly throughout. However, there are a number of reasons why this might not be the case. If one fly occludes another and they subsequently separate, it is difficult to be sure that the traces have not swapped. This seems unavoidable, given that it is difficult even for a human observer to accurately follow flies in this situation. Generally, whenever flies are close together there is a chance that traces will switch, e.g. due to one fly being temporarily undetected or a single fly giving rise to two blobs due to shadows and/or specularities. For this reason, the number of flies to be tracked simultaneously represents a trade-off between throughput and tracking accuracy; six was deemed to be the maximum number for which satisfactory performance was possible with the present system. The various parameters (e.g. pixel intensity difference threshold, minimum blob size, maximum matching distance) have been carefully tuned to minimise the frequency of such errors, but it is important to bear this source of noise in mind when considering the trace analysis, which we turn our attention to next.

## 4. Automated trace annotation

### 4.1: Introduction

Having produced traces of the flies' positions, the next step is to automatically annotate these traces to produce high-level descriptions of their behaviour. Automating this process is highly desirable, not least because manually annotating videos would be tedious and time-consuming. Moreover, if the program is written correctly, it will be more accurate than a human observer as it will apply exactly the same criteria in judging behaviour every time, and will not suffer from boredom, fatigue, or – worst of all – bias towards the desired outcome of the experiment.

We decided that the most appropriate representation of the flies' behaviour was as a sequence of discrete events, along the lines of: “went down at T-junction, turned around, went up at T-junction, fell down” etc. The alternative would have been a sequence explicitly predicated on time-steps, e.g. “walking right, walking right, stopped, stopped, stopped, walking up” etc. We chose the former for the following reasons:

- Events such as falls happen over a very short timescale, but should probably be treated as significant. It is not clear how to do this in the latter paradigm without having very short time-steps, in which case the representation is no longer high-level.
- We clearly want to record which way flies go at junctions, which makes more sense to represent as a single event rather than a behaviour with a certain duration.
- We intend to go on to probabilistically model flies' behaviour, and the former paradigm fits naturally into a Markov chain framework.

The decision of what behaviours to look for was partly determined by what we thought was interesting and potentially relevant to the flies' exit scores. For instance, we wanted to differentiate between falling and merely walking down, as even though both events might result in a fly being in the same location, their future behaviour may depend on which of these experiences they had. However, our choice was also limited by what could feasibly be detected. As a fly could appear as a blob as small as five pixels, it would be impossible to detect it turning on the spot or fighting with another male, for example. The events we selected are:

- Junction events
- Stops
- Falls
- U-turns

We shall discuss these in detail below. Note that walking is not treated as an event, but rather the absence of one.

### 4.2: Implementation

#### 4.2.1: Overview

The annotation is carried out by program written in Java. It takes saved traces one at a time and for each outputs a temporal series of events, as shown in box 4.2.1.1. The same trace is represented diagrammatically in figure 4.2.1.2. Every event's spatial and temporal position is recorded, with some additional information depending on the type of event. The program also outputs a matrix where each column represents a fly and the rows represent counts of instances of each event (although stops are

```

170   Went up at T-junction at (146,430)
188   U-turn vs 181 at (146,434)
204   Went down at T-junction at (142,469)
242   Went right at Y-junction at (194,510)
274   Went up at T-junction at (252,483)
327   Went right at Y-junction at (305,446)
361   Went up at T-junction at (366,423)
388   U-turn vs 377 at (370,413)
442   Stopped for 3.125s at (366,438)
443   U-turn vs 426 at (367,435)
515   Went right at Y-junction at (419,382)
554   Went up at T-junction at (484,363)
608   Went right at Y-junction at (529,325)
641   U-turn vs 634 at (546,324)
680   U-turn vs 669 at (529,325)
733   Went down at T-junction at (601,350)
769   Went right at Y-junction at (650,384)
794   U-turn vs 791 at (664,385)
822   Went down at Y-junction at (618,396)
848   U-turn vs 839 at (611,398)
874   Went up at Y-junction at (614,371)
894   U-turn vs 889 at (610,370)
1026  Stopped for 16.375s at (617,375)
1037  Went right at Y-junction at (649,386)
1095  Went up at T-junction at (724,372)
1119  Went right at Y-junction at (765,333)
1145  U-turn vs 1142 at (787,333)
1169  Went up at Y-junction at (727,319)
1179  Fell 11.7046995 at 70° to (729,324), proceeded up
1206  U-turn vs 1200 at (724,318)
1248  Went right at Y-junction at (762,333)
1273  U-turn vs 1268 at (788,332)
1285  U-turn vs 1281 at (794,329)
1315  U-turn vs 1306 at (796,329)
1328  U-turn vs 1317 at (795,336)
1346  Stopped for 3.375s at (805,332)
1389  Went down at T-junction at (842,353)
1433  Went right at Y-junction at (885,388)

```

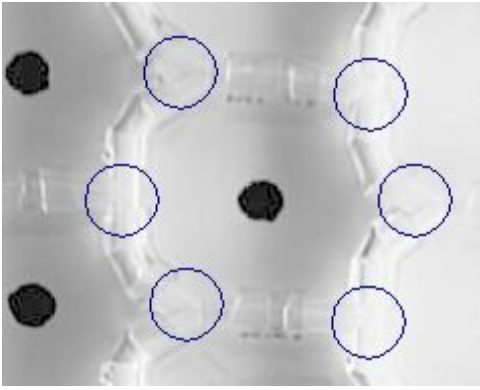
*Box 4.2.1.1: Annotation of a single c255 fly's trace. Numbers on the left are frame numbers, number in brackets are image co-ordinates of where the event took place. The same trace is shown in figure 4.2.1.2.*

treated differently, see below). This can be easily imported to a spreadsheet for statistical analysis.

Before annotating each trace, the program cleans it up by detecting short periods (up to 6 frames, i.e. 0.75s) where the fly was untracked and replacing them with a linear interpolation between the known positions. We also discard any events that take place left of the CoM of the leftmost spot (therefore including the first junction), as the traces for this region were found to be inaccurate. This is for two reasons: the close proximity of several flies causes traces to switch between them, and the fact that less than `nflies` flies may have entered the maze at this early stage means that blobs representing noise are more likely to be erroneously tracked.

In order to detect some of the events (see below) the program computes the fly's velocity at each frame based on the previous 1.5sec (i.e. 12 frames). If the trace was null at any point during this period (after cleanup) the velocity measurement is skipped. The length of this window was set through trial-and-





*Figure 4.2.2.1: Junction boundaries. The dark blue circles superimposed on this image depict the regions classed as junctions. Upon leaving one of these regions a fly will cause a junction event, assuming it leaves in a different way from that in which it came in. The radius of the circles is a constant proportion of the universal scale factor (see section 3.3.2).*

#### **4.2.3: Stops**

A stop is deemed to occur when a fly records no velocity of substantial magnitude for a period  $\geq 3$ sec. The threshold magnitude is defined in relation to the universal scale factor and framerate and corresponds to  $\sim 1.1$ mm/sec. There is no upper limit on the length of a single stop event. The duration of the stop is recorded, allowing the total time stopped to be calculated and expressed as a fraction of the time in the maze. This parameter is referred to as *Stop%*, and we consider it to be a better measurement of stopping behaviour than counting stop events.

#### **4.2.4: Falls**

A fall is defined as a fly moving more than a set distance (corresponding to maximum walking speed, set by trial and error to  $\sim 4.4$ mm) at an angle  $< \pi/4$  to the downward vector between subsequent frames. Note that falling out of a junction will cause both a junction and fall event to be registered.

As we hypothesise that how a fly proceeds after falling may depend on previous experience, we further analyse falls in terms of how the fly recovers from them. This is done by recording the first substantial velocity (greater than  $\sim 1.6$ mm/sec) observed after the fall. If this is at an angle  $< \pi/3$  to the upward vector the event is classified as a 'fall-up'; if the angle is  $< \pi/3$  to the downward vector it is a 'fall-down'. If neither of these is the case, the event is a 'fall-side'. This tends to occur when flies fall particularly far out of the bottom of a T-junction, and then proceed to the right. Note that if a fly falls twice in quick succession (though not consecutively; see section 4.2.6) the first will be treated as a 'fall-down'.

#### **4.2.5: U-turns**

At all times, the fly's last recorded substantial velocity is stored. If a new substantial velocity is recorded which is at an angle  $> 3\pi/4$  to the old one, a U-turn event is registered. Note that going from the top to bottom of a Y-junction is not a sharp enough turn to register as a U-turn. Movements made while falling are excluded from counting as U-turns, as is recovering from a fall. Thus, only if a fly resumes movement after a fall in the opposite direction to its movement before falling will the program report a U-turn event.

#### **4.2.6: Post-processing**

Some operations are applied to clean up the finished trace annotation. If two or more falls occur on consecutive frames, they are combined into a single fall event, as it is likely that the fly was recorded mid-fall on one (or more) frames. If two stops are separated by  $< 1$ sec, they are combined into a single stop, since the brief intervening movement was likely to have been due to noise.

## 5. Results

### 5.1: Event frequencies

#### 5.1.1: Method

Flies were tracked through the gravity maze in groups of 6. Trials with any obvious and significant tracking errors were discarded (e.g. a trace losing a fly, recording null for an extended period, and then picking up a different fly passing the same position). Trials lasted a maximum of 15 minutes; the traces of any flies which had failed to finish in this time were discarded (if this was more than 2 out of 6, the whole trial was discarded). In total, between 21 and 29 flies were tracked for each of the ten strains; fig 5.1.1.1 shows the mean exit scores. The traces were automatically annotated as described in chapter 4.

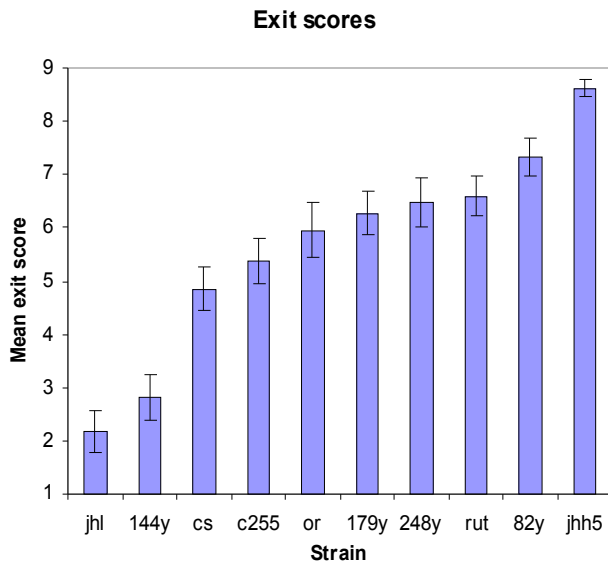


Figure 5.1.1.1: Mean exit scores obtained in the video tracking experiments. Bars represent  $\pm 1SE$ , treating each fly as an independent measurement. *jhl*:  $n=21$ , *144y*:  $n=27$ , *cs*:  $n=28$ , *c255*:  $n=24$ , *or*:  $n=22$ , *179y*:  $n=29$ , *248y*:  $n=23$ , *rut*:  $n=22$ , *82y*:  $n=22$ , *jhl*:  $n=24$ .

The most basic analysis we can carry out is to calculate the frequency of each type of event and look for strain-dependent differences. Figure 5.1.1.2 shows the results. We have ignored Y-junction events because these are rather difficult to interpret owing to the fact that there are no Y-junctions along the top or bottom edges of the maze, meaning that extreme high and low strains encounter them less frequently than those that exit around the middle. Furthermore, the anti-phototactic junction events (T-left, Y-up and Y-down) occur too seldom to obtain reliable estimates of their frequency.

One-way ANOVAs reveal highly significant effects of strain on all of the parameters displayed in fig 5.1.1.2. *Post hoc* Duncan's tests can therefore be used to examine the nature of these effects (see table 5.1.1.3). Additionally, figure 5.1.1.4 shows how the various strains tend to recover after falls.

#### 5.1.2: Discussion

In terms of exit score, we see some puzzling results. *179y* scores considerably lower than we would expect from Armstrong *et al's* (2005) findings. (*248y*, on the other hand, was only identified as having a modest anti-gravitactic phenotype.) *Rut*, meanwhile, scores considerably higher than it did in control experiments conducted in the same maze just a week or two earlier. These observations are difficult to account for, and perhaps simply underscore the difficulty of controlling behavioural experiments;

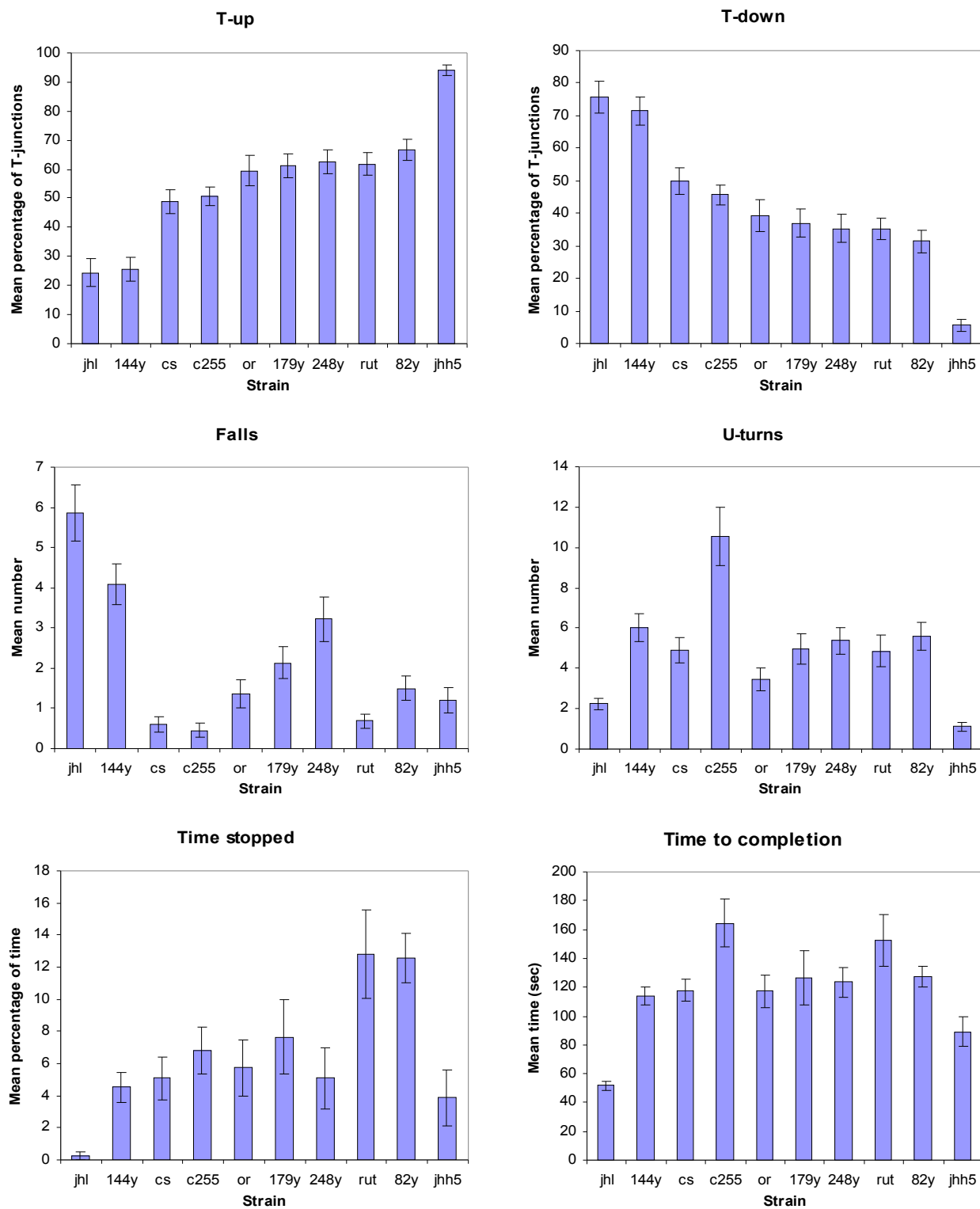


Figure 5.1.1.2: Basic event-based comparison of strains, which are shown in ascending order of mean exit score. (a) and (b) show the mean percentage of T-junctions at which flies went up and down, respectively. Note that these typically add to <100% due to anti-phototactic T-left events. (c) displays the mean number of U-turns and (d) number of falls. (e) gives the amount of the time spent stopped, expressed as a percentage of the time taken from leaving the first column 2 T-junction encountered to reaching an exit, which is shown in (f). Bars represent  $\pm 1SE$ .

	jhh5	82y	rut	248y	179y	or	C255	cs	144y
jhh5	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.268
82y	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	
rut	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	
248y	0.000	0.000	0.005	0.008	0.019	0.065	0.358		
179y	0.000	0.001	0.053	0.073	0.132	0.304			
or	0.000	0.029	0.310	0.385	0.568				
C255	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	
cs	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	
144y	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	

**Exit scores**,  $F(9,232) = 23.71$ ,  $p \approx 0$ .

	jhh5	82y	248y	rut	179y	or	C255	cs	144y
jhh5	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.806
82y	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	
248y	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	
rut	0.000	0.004	0.028	0.035	0.042	0.069	0.740		
179y	0.000	0.010	0.055	0.066	0.076	0.112			
or	0.000	0.263	0.622	0.698	0.777				
C255	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	
cs	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	
144y	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	

**T-up%**,  $F(9,232) = 25.83$ ,  $p \approx 0$ .

	jhl	144y	cs	C255	or	179y	rut	248y	82y
jhl	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
144y	0.000	0.000	0.003	0.022	0.213	0.375	0.488	0.510	
cs	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	
C255	0.000	0.000	0.017	0.088	0.506	0.778	0.990		
or	0.000	0.000	0.019	0.095	0.517	0.784			
179y	0.000	0.000	0.031	0.135	0.667				
rut	0.000	0.000	0.071	0.248					
248y	0.000	0.000	0.454						
82y	0.000	0.000	0.000	0.000					
144y	0.463								

**T-down%**,  $F(9,232) = 25.92$ ,  $p \approx 0$ .

	C255	144y	82y	248y	179y	cs	rut	or	jhl
C255	0.000	0.000	0.000	0.000	0.001	0.001	0.001	0.034	0.287
144y	0.000	0.001	0.004	0.006	0.018	0.018	0.016	0.244	
82y	0.000	0.031	0.073	0.100	0.191	0.196	0.177		
248y	0.000	0.337	0.546	0.652	0.928	0.978			
179y	0.000	0.339	0.550	0.657	0.945				
cs	0.000	0.357	0.577	0.684					
rut	0.000	0.564	0.849						
or	0.000	0.669							
jhl	0.000	0.000							

**U-turns**,  $F(9,232) = 11.49$ ,  $p \approx 0$ .

	jhl	144y	248y	179y	82y	or	jhh5	rut	cs
jhl	0.000	0.000	0.000	0.006	0.099	0.146	0.219	0.705	0.787
144y	0.000	0.000	0.000	0.012	0.152	0.215	0.307	0.892	
248y	0.000	0.000	0.000	0.016	0.179	0.246	0.339		
179y	0.000	0.000	0.001	0.125	0.622	0.778			
82y	0.000	0.000	0.001	0.186	0.804				
or	0.000	0.000	0.001	0.186	0.804				
jhh5	0.000	0.000	0.003	0.247					
rut	0.000	0.001	0.050						
cs	0.000	0.120							
144y	0.001								

**Falls**,  $F(9,232) = 19.62$ ,  $p \approx 0$ .

	rut	82y	179y	C255	or	248y	cs	144y	jhh5
rut	0.000	0.000	0.007	0.016	0.044	0.073	0.067	0.095	0.134
82y	0.001	0.001	0.187	0.295	0.498	0.649	0.642	0.791	
179y	0.002	0.003	0.269	0.405	0.645	0.820	0.815		
C255	0.004	0.005	0.357	0.518	0.793	0.991			
or	0.004	0.004	0.348	0.507	0.787				
248y	0.007	0.008	0.470	0.659					
cs	0.021	0.023	0.740						
144y	0.041	0.041							
jhh5	0.915								

**Stop%**,  $F(9,232) = 4.67$ ,  $p < 0.0001$ .

	C255	rut	82y	179y	248y	cs	or	144y	jhh5
C255	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.001	0.031
rut	0.000	0.001	0.057	0.061	0.079	0.133	0.127	0.157	
82y	0.010	0.054	0.505	0.532	0.618	0.824	0.840		
179y	0.017	0.077	0.614	0.645	0.740	0.971			
248y	0.017	0.077	0.626	0.655	0.749				
cs	0.034	0.131	0.834	0.873					
or	0.044	0.157	0.948						
144y	0.043	0.151							
jhh5	0.491								

**Time**,  $F(9,232) = 5.98$ ,  $p \approx 0$ .

*Table 5.1.1.3 Tables showing pairwise comparisons of strains on the various parameters. Entries are p-values computed by Duncan's tests. Colour coding is as follows:  $p < 0.05$ ,  $p < 0.01$ ,  $p < 0.001$ . Strains are listed in order of their score on each parameter. Captions give the result of the corresponding one-way ANOVA.*

factors as subtle as daily temperature variation could be to blame. These discrepancies could also be caused by the low numbers of replicates. Although we tracked ~24 flies for each strain, as discussed in section 2.3 it is perhaps invalid to treat them as independent measurements as the flies were put in the maze in groups. Only 4-5 separate trials were conducted for each strain.

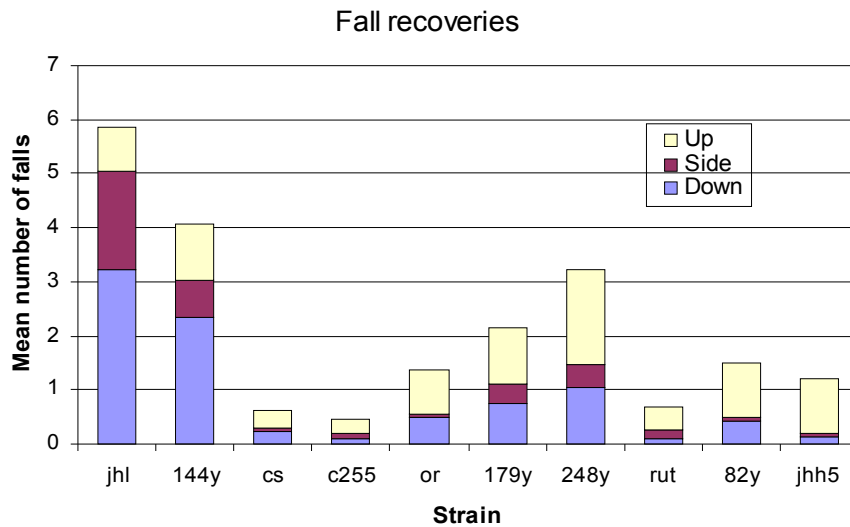


Figure 5.1.1.3: Breakdown of fall events in terms of recoveries. Note that these are absolute numbers rather than proportions (as with T-up and T-down) since many flies never fall for the duration of a trial.

A number of observations worthy of comment emerge from the behavioural analysis. Firstly, there are very robust differences between many strains, which is obviously encouraging. However, note that none of the differences between, for instance, *cs* and *or* are significant. This is probably due to a combination of similar behaviour between the strains, a low number of flies (~24 per strain) and noise in the form of tracking and/or annotation errors.

Unsurprisingly, a close association exists between T-up/T-down frequency and exit score. Similarly, the low strains fall most frequently, although note how infrequently *cs* and *c255* fall. Fall recoveries are as one might expect: high strains tend to climb up after a fall, low ones continue downward. *c255* U-turns markedly more frequently than any other strain. This may be due to visual defects rendering it unable to properly sense the light gradient. *Gap1* (the disrupted gene in *c255*) is involved in the development of the eye; mutants' eyes have occasional fused ommatidia, an excess of photoreceptors in each ommatidium, and abnormal distribution of pigment cells (Ashley & Katz, 1994), although the extent to which these defects affect their phototactic performance is unknown.

*Rut* and *82y* spend much more time stopped than the other strains. In the case of *rut*, this could represent a locomotor defect. Release of acetylcholine at the neuromuscular junction in response to tetanic stimulation is depressed in *rut* mutants (Kuromi & Kidokoro, 2000). *82y* is more puzzling; it has been previously found to be hyperactive (D.A. Baker, personal communication). However, this observation was based on its frequency of crossing a light gate in the middle of a small corridor, so it could be that *82y* is sporadically hyperactive with periods of quiescence. Its relatively short mean completion time (c.f. *rut*) could support this account.

*Jhl* and *jhh5* both appear to be highly motivated, as evidenced by their low frequency of U-turns and stops, and their quick completion of the maze. Quite why this should be the case is unclear, as they were not selected for speed of gravity maze completion, only exit score. These findings could perhaps reflect a higher degree of overall genetic fitness.

## 5.2: Correlations of events with exit scores

We shall now investigate the question of what factors determine an individual fly's exit score. Obviously, its choices of direction at T-junctions will be a large determinant. However, it could also be the case, for example, that prolonged periods of stopping could predict a low eventual exit point. We can address this issue by calculating the correlation between the various behavioural measures and exit score. Table 5.2.1 shows the results, for each strain individually and all flies considered together.

	T-up%	T-dn%	Falls	Stop%	U-turn	Time
All	0.939	-0.931	-0.408	0.122	-0.042	0.114
jhh5	0.787	-0.821	-0.213	-0.058	0.121	-0.162
82y	0.703	-0.755	-0.563	-0.427	-0.092	-0.200
rut	0.800	-0.816	0.259	0.071	0.177	0.306
248y	0.905	-0.913	-0.258	0.136	0.112	0.229
179y	0.926	-0.943	-0.234	0.066	0.158	0.085
or	0.914	-0.896	0.065	-0.151	0.100	-0.161
c255	0.819	-0.776	-0.169	-0.028	-0.115	-0.139
cs	0.955	-0.939	0.185	-0.212	-0.051	-0.279
144y	0.917	-0.896	-0.174	0.078	0.174	0.354
jhl	0.867	-0.867	-0.612	0.237	-0.210	0.004

Table 5.2.1: Table of correlation coefficients (Pearson's  $r$ ) between exit score and a selection of behavioural parameters. Significance levels are denoted by colour coding:  $p < 0.05$ ,  $p < 0.01$ ,  $p < 0.001$ . Note that no correction is made for the multiple comparisons.

As one would expect, for all strains T-up% and T-down% are highly positively and negatively correlated with exit score, respectively. Other than that, few effects are seen. This is probably due to the small samples used, meaning that only the largest effects will show up. Overall, fall frequency is negatively correlated with exit score, but this may simply reflect that the lowest-finishing strains tend to fall the most (see fig 5.1.1.2), without telling us anything about the behaviour of individual flies of other strains. It is curious that *jhl* shows a correlation between falling and exit score, while the other very low strain, *144y*, does not. This could be explained by *144y* flies making more of an effort to recover from falls; they proceed upwards after 25.5% of falls, whereas *jhl* do so only 13.8% of the time. (We can consider fall-downs and fall-sides roughly equivalent as they both tend to occur below T-junctions; the distinction merely reflects how far the fly falls.) A similar argument can be applied to the fact that *82y* exhibits a correlation but *jhh5* does not: *jhh5* proceed upwards following 82.8% of falls, compared to only 66.7% for *82y*. These effects may only be noticeable in the extreme high and low strains because their stereotyped behaviour leads to ceiling and floor effects in the exit scores, making any deviations from the normal pattern (due to falls in *jhh5* and *82y*, and not falling in *jhl* and *144y*) particularly conspicuous.

It is unclear how to interpret the link between stopping and low exit scores in *82y*. One should bear in mind that we have performed many tests and this correlation is not particularly strong, so may represent a chance occurrence. However, if we treat it as a genuine effect, it could reflect variability in the general fitness of *82y* individuals, with those stopping most frequently also being least able to climb. The disrupted gene, *off-track*, is involved in nervous system development, and it has been observed that *82y* adults can exhibit rather variable degrees of neuroanatomical abnormality, particularly in the mushroom bodies (D. A. Baker, personal communication).

## 5.3: Spatial analysis

### 5.3.1: Method

As previous work had suggested that the flies' behaviour is not uniform throughout the maze, we were keen to assess whether the behavioural parameters recorded differed according to the flies' spatial positions. Due to the small number of replicates, we could not perform a high-resolution analysis. Therefore we simply compare the behaviour in the first half of the maze (hereafter 'early') to that in the latter half ('late'). More accurately, all events occurring right of the leftmost spot and left of the fourth column of spots (i.e. T-junctions 2-4, inclusive) are counted as early; those right of the fifth column and left of the eighth (junctions 6-8) are late (see fig 5.3.1.1). Note that all events occurring inside the zone count, even if the fly leaves and then returns due to backtracking.

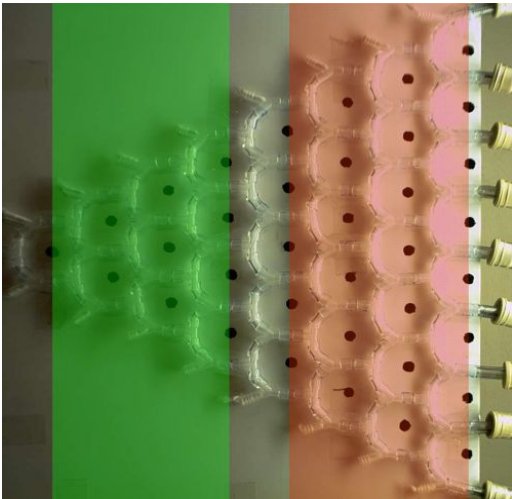


Figure 5.3.1.1: Spatial analysis. The 'early' zone is highlighted in green, the 'late' in red. Note the 'buffer zone' between them.

Histograms of the behavioural parameters are given in figure 5.3.1.2, and table 5.3.1.3 shows the results of carrying out t-tests between the early and late parameter for each strain. We cannot perform t-tests for fall-up and fall-down, as it is most appropriate to express these as a proportion of total falls, and many flies never fall in one or more zones, meaning that the proportion is undefined. We therefore use chi-squared tests to compare the number of fall-ups and fall-downs in each zone, ignoring fall-sides. (If one or more of the expected values is  $<5$ , we instead use a Fisher exact probability test, as chi-squared becomes unreliable in such circumstances.) As each case represents a fall event, rather than a fly, those flies that fall more often will contribute more to the result. Results are given in table 5.3.1.4.

### 5.3.2: Discussion

The most striking result is that many of the strains stop more frequently in the late section of the maze. This difference is significant in the cases of *jhh5*, *179y*, or *c255*, and for many other lines the same trend is apparent; it is thus reasonable to suggest that with more data significance could be achieved. However, the clear exceptions are *rut* and to a slightly lesser extent *82y*, which stop frequently in both sections. Interestingly, both *rut* and *82y* are known to have learning defects. It could therefore be that normal flies become less motivated to explore as the novelty of the repetitive maze wears off, causing the increase in stoppage time. The memory mutants, on the other hand, 'forget' their past experience in the maze. If this were the case, however, one might expect the memory mutants to exhibit consistently low stopping, rather than high.

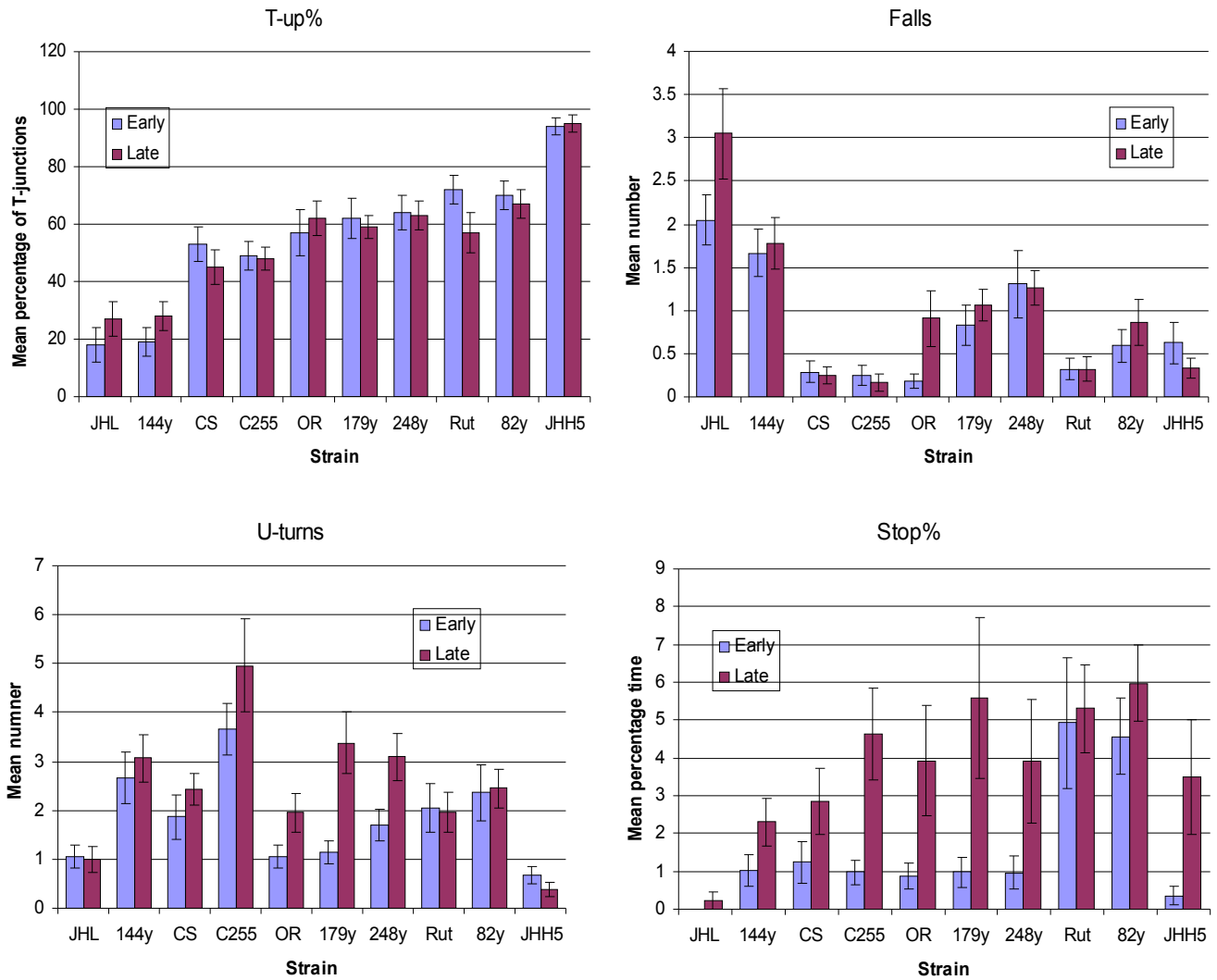


Figure 5.3.1.2: Early vs. late behaviours. T-up% refers to the percentage of T-junctions encountered in that zone at which the fly went up. T-down% is excluded as it is approximately the opposite of T-up% (see fig 5.1.1.2). Stop% is the percentage of time taken to complete the whole maze that the fly spent stationary in that zone. Bars represent  $\pm 1SE$ .

	T-up%	Fall	Stop%	U-turn
Jhh5	0.249	1.127	2.070	1.273
82y	0.421	0.866	1.003	0.135
Rut	1.862	0.000	0.179	0.142
248y	0.163	0.101	1.766	2.439
179y	0.370	0.834	2.166	3.356
Or	0.453	2.231	2.098	2.038
C255	0.256	0.569	2.966	1.218
Cs	0.917	0.225	1.562	1.049
144y	1.349	0.275	1.732	0.587
Jhl	1.118	1.720	1.000	0.137

Figure 5.3.1.3: t-values obtained comparing early and late parameters for each strain. Significance levels are denoted by colour coding:  $p < 0.05$ ,  $p < 0.01$ . Note that no correction is made for the multiple comparisons.

	N	Chi-sq	P	
Jhh5	22	n/a	0.788	Table 5.3.1.4: Early vs. late fall recoveries. Chi-square (df=1) and p-values for each strain, comparing early/late vs. fall-up/fall-down absolute frequencies. Entries of 'n/a' indicate that a Fisher exact probability test was used instead of chi-square. N is the total number of fall events contributing to each test.
82y	30	n/a	0.618	
Rut	11	n/a	0.545	
248y	52	0.000	1.000	
179y	48	0.022	0.882	
Or	23	n/a	0.822	
C255	8	n/a	0.214	
Cs	13	n/a	0.653	
144y	76	0.898	0.343	
Jhl	75	0.012	0.913	

This hypothesis is interesting, but is merely a suggestion at this stage. Many further experiments would be required to rule out alternative explanations for the effect. For instance, the increased stopping of healthy strains in the latter half of the maze could be due to physical fatigue rather than flagging motivation. *Rut* and *82y*'s consistently high stopping could therefore indicate that they become fatigued much more quickly. Indeed, the observation that neuromuscular ACh release during high frequency stimulation is depressed in *rut* mutants (Kuromi & Kidokoro, 2000) could lend support to this view. Alternatively, the increased ambient light in the latter half of the maze, rather than any kind of experience effect, could be causing the difference in normal flies. Perhaps it is sufficiently bright in the late portion of the maze that flies' attraction to the light source decreases, i.e. their brightness perception begins to saturate.

A similar pattern to that of stopping time is observed with U-turn frequency, although it is considerably less clear. Many of the arguments made above about stopping can equally be applied here. However, the active nature of U-turning compared to stopping may suggest that the effect is due to motivational factors rather than simple fatigue.

The finding that *or* (along with *jhl*, though the effect is non-significant) falls more frequently in the latter section is perhaps best explained as a consequence of fatigue, though it is not clear why a supposedly healthy wild-type should exhibit this effect most strongly. Turning our attention to the analysis of fall recoveries, we see no significant effects whatsoever. Although this could reflect a lack of data (especially in the cases of *rut*, *c255* and *cs*), the fact that none of the results even approach significance indicates that fall recoveries do not change as a function of position in the maze. This suggests that we should reject the hypothesis that repeated falls make a fly more likely to progress downwards from subsequent falls.

This analysis has shown that behaviour does change as flies progress through the maze. However, the finding that neither T-junction choices nor direction of fall recovery changes significantly for any strain suggests that these changes have little impact on exit score. We should however be cautious not to jump to the conclusion that gravity responses are unaffected by experience, as this experiment possibly lacked both the spatial resolution and quantity of data to uncover more subtle effects.

## 6. Modelling behaviour

### 6.1: Introduction

Based on the data we have collected we can attempt to create a probabilistic model for each stain of fly, which will predict what it is most likely to do next given its current situation. Such a model allows us to assess how well we have characterised the flies' behaviour, as we can run virtual flies through a simulated maze and compare their performance to the real thing.

We model the a fly as a Markov process. A Markov model can be thought of as giving the probabilities of various actions given the current state, and the transition probabilities for various potential next states given that action and the current state. The actions are the behavioural events such as T-up, U-turn, etc. The state is given by the fly's *surroundings* and recent behaviours. *Surroundings* describes the fly's current position in the maze and may be one of: *T-junction*, *Y-junction* or *tube*. The surroundings constrain the actions that are possible:

$$p(\text{action}|\text{surroundings} = T\text{junction}) = \begin{cases} a \text{ if } \text{action} \in [tUp, tDown, tLeft], 0 \leq a \leq 1 \\ 0 \text{ otherwise} \end{cases}$$

and so on. As well as the current surroundings, the probability distribution is based on the previous two events:

$$p(\text{action}|\text{surroundings}, \text{event}_{t-1}, \text{event}_{t-2})$$

Thus, we are using an  $n$ -step Markov chain with  $n=2$ . This choice of  $n$  represents a trade-off: a high  $n$  means a more detailed model, but the state space grows exponentially with  $n$  so much more data is required to properly estimate the probability distribution.

A central assumption of the Markov model approach is that the problem has the *Markov property*: the current state predicts future states just as well as the the entire history of states leading to this one, i.e. the process has no memory (Sutton & Barto, 1998). This is a very strong assumption, and in this case it is clearly invalid; flies do have memories that last longer than the previous two events. Nonetheless, the approach can still be used as an approximation.

### 6.2: Simulating the maze

Before describing how the model is actually built, we shall explain how state transitions are dealt with. Because the topology of the maze is known, the state transitions can actually be made deterministic with a little care. A dummy action, *proceed*, is introduced into the event sequences output by the annotation program. It is inserted immediately before every junction event, and denotes that the fly continued to the junction ahead without incident. When simulating behaviour, a simple model of the maze that maintains the fly's current position and heading is employed. Thus this model can predict which junction *proceeding* will lead the fly to at any given time. Similarly, once a fly is at a junction, its choice of direction is used to tell the model which section of tube it is in, i.e. where the next *proceed* event would take it. U-turns cause the model to switch the previous and next junctions, as can fall-ups and fall-downs, dependent on the fly's heading prior to the fall. Stops have no effect on position or heading. This process is summarised in fig 6.2.1. The fly is initialised in the leftmost T-junction, and any attempt to proceed to the left results in it looping back to that junction, preventing it from escaping

through the entrance. When a proceed event takes the fly out of the right side of the maze the model terminates, outputting an exit score.

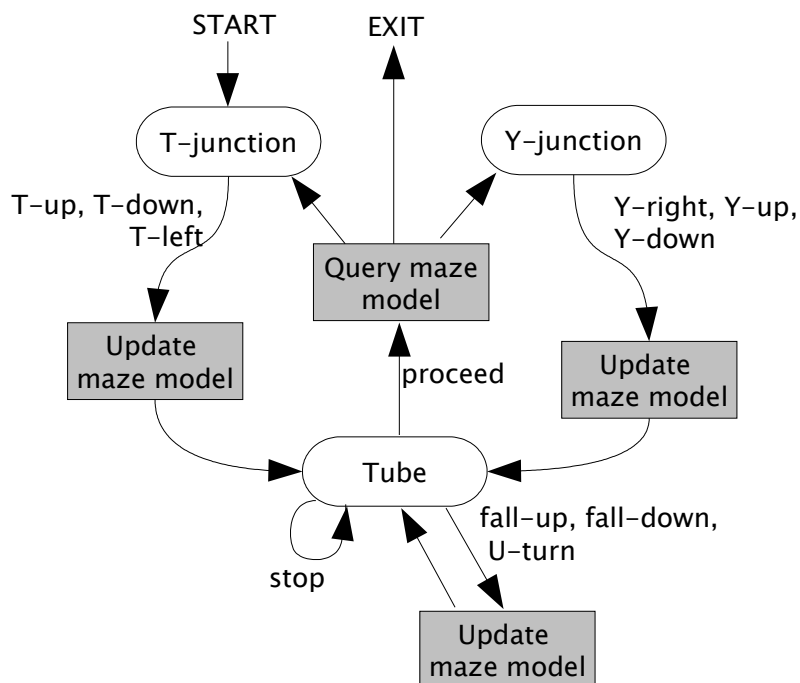


Figure 6.2.1: State transition diagram. Rounded boxes represent possible surroundings, italics represent events, and the shaded boxes times when data is passed to or from the maze model. See text for full explanation.

Some minor modifications needed to be made to the annotation program for this scheme to work. Firstly, U-turns occurring inside junction areas (see fig 4.2.2.1) were ignored. This is because their effect depends on whether the fly U-turned before or after changing direction in the junction, which we cannot assess. Based on informal observation, most commonly a U-turn inside a junction would be caused by the fly starting to go one way at the junction (but not continuing far enough to leave it) then choosing the other unexplored route, rather than leaving the way it came in. This would therefore be treated as a single junction event.

Fall-sides were scrapped, with any continuation below or on the horizontal being treated as a fall-down, and anything above a fall-up, due to the difficulties of predicting a fly's direction following a fall-side. Also, U-turns consisting of opposite directions of movement before and after a fall were no longer recognised, as this not only has unpredictable effects in the framework described, but is redundant given the fall-up/fall-down distinction.

### 6.3: Constructing the model

The model is built by iterating through a fly's annotation, with *proceed* events added. The *surrounding* variable is determined by the event type: T-junction events can only take place at T-junctions, Y junction events at Y-junctions, and all other events only in tubes. The two previous events are specified, obviously enough, by the preceding two events. However, *proceed* events are ignored (i.e. they can only be actions, not part of the state description) as they give no information; they simply come before every junction event. A count is kept of the number of times each possible ( $event_{t-2}$ ,  $event_{t-1}$ , *surroundings*, *action*) 4-tuple is encountered. This process is repeated for all the flies of that strain. Once this is completed, the probability of an action given the state can be calculated as follows:

$$p(\text{action}|\text{state}) = \frac{n(\text{action}|\text{state})}{\sum_{a \in \text{actions}} n(a|\text{state})}$$

where *state* is (*event<sub>t-2</sub>*, *event<sub>t-1</sub>*, *surroundings*) and *n(a|s)* is the count of occurrences of action *a* in state *s*.

In fact, during this process a 1-step and 0-step model are also being built in exactly the same way but using fewer previous events in the state description (none in the 0-step case, which simply represents the prior probability of each action in the corresponding surrounding). This is done for two reasons:

- At the beginning of the simulation a 2-step history is unavailable. One could use special dummy events to code for this situation, but this would increase the state space and the probabilities for states involving these would be poorly estimated due to their infrequency.
- Sometimes a simulated fly may find itself in a state that is only represented a few times in the training data, and therefore has unreliable probabilities. In the worst case, the state might never have been seen before. In such a situation, it is desirable to use probabilities for a less specific and therefore more frequently encountered state. We have hence implemented a 'back-off' algorithm that will revert to an (n-1)-step Markov model if the total number of observations upon which any probability is based is <5.

## 6.4: Results and discussion

A Markov model was produced and 500 flies simulated in a Monte Carlo fashion for each strain. Note that this is many more trails than we have for the real flies; the high number was chosen to obtain accurate means. Their event frequencies were recorded just as before, although note that time to completion and proportion of time stopped could not be obtained as the model does not represent the passage of time. Histograms are shown in fig 6.4.1. Generally, the model matches the data fairly well. However, there are a number of areas in which there is clearly some disparity, most noticeably the exit scores for *rut* and *82y*, and the U-turn frequency for *jhh5*. We shall consider the possible reasons for this inaccuracy.

Firstly, section 5.3 showed that flies do behave differently depending on their position in the maze. A model predicated only on their immediate history will be unable to capture these patterns. It may be no coincidence that *rut* exhibits the largest difference in T-up% between early and late in the maze (fig 5.3.1.2, though the effect is non-significant) and is the strain whose exit score is predicted least accurately by the model. A related problem is that behaviour at the first junction is not recorded, but the model still attempts to simulate it. Flies might behave quite atypically at the first junction, as this is the time when interactions with other flies are likely to be most frequent. For instance, *jhh5* flies can sometimes be seen falling at the first junction as a result of jostling with each other, but this is not captured by the model. This could explain the unrealistically high mean exit score for this strain: 8.90 out of a possible 9.

Another possible weakness is that probabilities are based on the raw frequencies of events summed over all flies. Thus flies which perform more events may have a disproportionately large effect on the model. This could introduce bias, as those flies responsible for the most events tend to be the least 'directed' ones, i.e. those which frequently stop, U-turn, and take long routes through the maze. This could possibly explain the model's tendency to predict more frequent falls (and U-turns, though the trend is less obvious) than are observed in the real data. This problem could perhaps be alleviated by normalising each fly's contributions to the model according to the total number of events it performed.

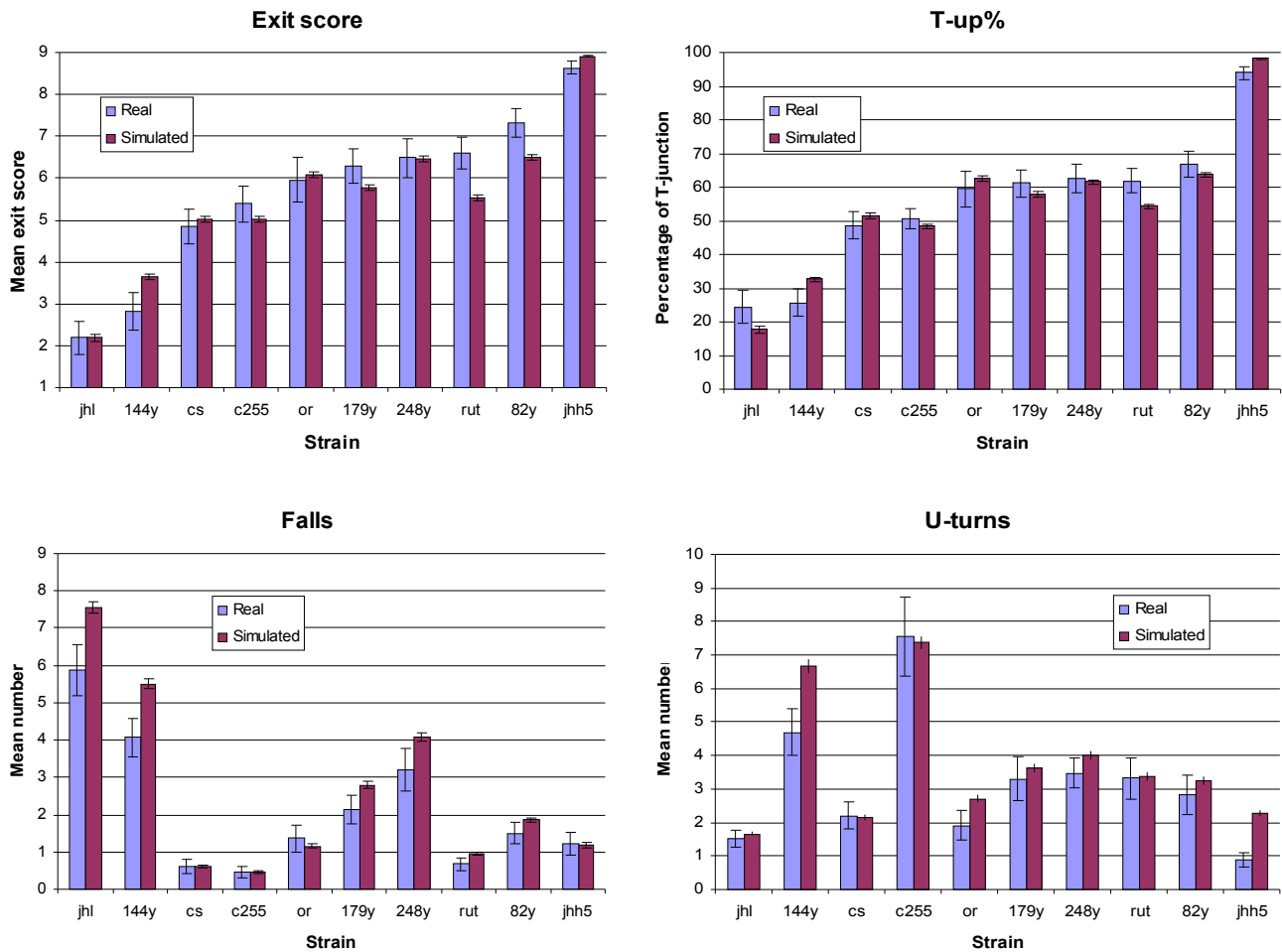


Figure 6.4.1: Mean exit scores and event frequencies for the video tracking data and 500 simulated runs (per strain) using the Markov model. Note that U-turn frequencies for the video tracking data have been recalculated using the same criteria applied to build the model, i.e. excluding those in junctions or interrupted by a fall.

However, this might simply introduce new biases, for example flies taking the top or bottom edge of the maze avoid Y-junctions and therefore register fewer events than those taking less extreme routes.

It is also possible that our model is missing crucial details. It ignores the direction from which a fly arrived at a junction, which is very likely to affect the direction in which it leaves. As mentioned above, it also ignores interactions between flies. This area has been rather neglected in the project as a whole, a point which we shall return to in the discussion.

As with all modelling, there is a danger that in making the problem tractable we introduce unwarranted and unhelpful abstractions. For example, we could fit a mixture of Gaussians to each strain's exit profile, and claim that we had an accurate, generative model of gravitaxis. However, this would tell us almost nothing about the system we are supposedly modelling. The model described here is rather more detailed than that, producing sequences of behavioural events which appear to match reasonably closely with biological data. However, it is unable to model the temporal nature of the behaviour, abstracting over factors such as stop duration and walking speed.

In fact, this simplification could be another potential source of inaccuracy. It is reasonable to assume that a fall two seconds ago is more likely to affect a fly's current behaviour than one 30 seconds ago, but our model makes no distinction between the two. It is not clear how to address this within the Markov framework, but a relatively simple possibility would be to add a 'no event' event to the behavioural annotation, corresponding to, say, 15 seconds passing without any other event occurring. Old events would thus get pushed out of the state representation by the passage of time.



## 7. Discussion

### 7.1: Conclusions

We have presented a computer vision system that allow multiple flies to be tracked in a gravity maze paradigm, the dominant assay for *Drosophila* gravitaxis since the seminal studies of Hirsch and collaborators in the 1960s. We describe software that automatically analyses the behaviour of each fly, detecting events such as junction decisions, falls and U-turns. The development of these methods has made it possible to obtain quantitative descriptions of the behaviours that occur inside the maze, and thus contribute to our quantification of gravitactic instinct, for the first time.

We have characterised the behaviour of a number of strains, including two wild types, artificially selected positive and negative gravitactic lines, mutants identified as having a specific gravitactic phenotype, and the learning mutant *rutabaga*. Despite analysing only ~24 flies of each strain, we were able to detect robust behavioural differences between many of the lines. We conducted a correlation analysis and found that for many strains the only predictor of exit score was up/down choices at T-junctions, but for some frequent falling was also linked to low exit scores.

Based on a previous preliminary study we hypothesised that flies' behaviour may change as they progress through the maze. In particular, it was thought that repeated falls would make them less likely to attempt to climb in future. A spatial analysis of events in the two halves of the maze supported the former hypothesis for many strains, but no evidence was found for the latter.

Markov models for each strain were constructed based on the behavioural data, allowing Monte Carlo simulations of the gravity maze to be run. The resultant exit profiles represented an approximate but imperfect match with the observed data. Various possible reasons for this discrepancy, such as ignoring junction entry direction and the time elapsed between events, are discussed in section 6.4.

Some interesting and unexpected results worthy of further investigation have emerged. Most notably, the spatial analysis revealed that flies generally stop (and perhaps U-turn) more often in the latter half of the maze. However, *rut* and *82y* fail to exhibit this pattern. As both strains have a memory defect, we tentatively hypothesise that the normal behaviour may reflect reduced motivation due to the decreasing novelty of the repetitive maze environment.

### 7.2: Evaluation

Due to the short timescale of this project (14 weeks), shortcuts had to be taken to cover the ground we have. In particular, the software was developed in a rather *ad hoc* fashion. Testing was carried out by running the programs on a few examples and subjectively judging how well they tracked flies or annotated behaviour. Formal tests of accuracy, i.e. comparing the programs' performance to that of manual annotation by experts, would therefore be very useful. Not only would these allow us to quantify the system's reliability, but also to identify its weakest points and improve them. There are many potential refinements that could be made. For instance, the maximum distance for assigning blobs to flies (see section 3.3.4) could be made adaptive, i.e. start at just a few pixels and grow outwards with every frame in which that fly was undetected. The centre of the matching region could also move according to the last record of the fly's velocity, thus projecting its likely position. The annotation program could be modified to look for tracking errors (e.g. sudden jumps back and forth caused by briefly tracking noise) and repair them by interpolating between more reliable readings. However, these

possibilities were not implemented as performance was deemed 'good enough'; it was better to spend 4 weeks building a system that could track 6 flies simultaneously than 6 weeks on one that could track 8.

In addition to quantifying and improving accuracy, it would be desirable to do likewise for robustness. We could assess the extent to which the system can tolerate different lighting conditions, different cameras and framerates, and even mazes of different sizes or made from different materials. These measures would potentially allow the software to be used by other laboratories with minimal reconfiguration.

The disagreements between mean exit scores from the video tracking maze and previously reported data, or worse still our own control experiments in the same maze (compare *rut* in figs 2.3.3 and 5.1.1.1), are cause for concern. These perhaps highlight the dangers of treating each fly as an independent observation when they are interacting in the maze. If, for the sake of argument, flies are inclined to follow each other, then the actions of the first fly out of the entrance chamber could potentially skew the results of the entire trial.

Few firm findings have emerged from the behavioural analysis. However, this is perhaps to be expected. Apart from the preliminary study discussed in section 1.3, no attempts had previously been made to investigate behaviour inside the gravity maze, so we had few specific hypotheses at the outset. The purpose of the project was more to characterise the behaviours exhibited by various strains, making it necessarily exploratory. It is therefore encouraging that we have made observations that warrant further investigation. Simply repeating the experiments discussed here with more flies may well yield more results, as evidenced by the various visible trends that failed to reach significance.

Although the Markov modelling approach we adopted could achieve a rough match with observed behaviour, there are several respects in which its predictions are at odds with the data. Thus, although we have been able to describe gravitaxis behaviour in considerable detail, unfortunately we have not characterised it sufficiently well to accurately reproduce it.

### 7.3: Future work

In this section we shall discuss opportunities for further research, starting with fairly straightforward extensions to this study and moving towards more ambitious and open-ended projects.

#### 7.3.1: Improving visualisation facilities

The tracking and annotation systems make it possible to produce large amounts of data very quickly. If researchers are to intuitively grasp the patterns evident in this mass of data, good visualisation techniques are essential. At present, one can view individual traces with or without symbols denoting behavioural events (see fig 4.2.1.2) and also view several traces from the same trial. However, to visualise all the data for a particular strain, or to compare between strains, it is necessary to combine data from several different trials. This is a non-trivial problem, as the relative positions of the camera and maze differ slightly from trial to trial. It would therefore be necessary to map positions onto some standardised maze template first, which could be achieved thanks to the recording of dot/junction positions with each trace.

Once such functionality was implemented, one would be able to split the maze into sectors and produce a density plot of the average length of time spent, or the average frequency of particular behavioural events, in each one. Combined with more data collection, this would give much more detailed

information than the rudimentary spatial analysis described in section 5.3.1; one might for instance be able to see exactly where fatigue begins to set in for each strain. Visualisation facilities of this kind would also be very useful for displaying results of simulations, thereby giving a clearer indication of exactly where deviations from the biological data were occurring.

### **7.3.2: Including time information**

The representation of behaviour as a discrete series of events is convenient for Markov modelling. However, as we have noted, this overlooks factors such as walking speed and duration of stops. Some strains move considerably faster than others, as shown by the mean completion times (fig 5.1.1.2). It would be desirable to represent this information in our model, as walking speed may well alter as a function of experience, or have an effect on subsequent behaviours. Furthermore, it may give us an insight into motivation levels, although one would need to control carefully for differing locomotor capabilities between strains.

A first approach we could take to this issue would be to discretise speeds (e.g. slow, medium and fast) and record the average walking speed between consecutive pairs of events. Alternatively, we could attempt to detect changes in speed as they happen. In the latter case, it would be appropriate to adopt a buffer zone system as in Martin (2004) to prevent minor fluctuations around the thresholds from being counted. In either case, these 'walking events' could be easily accommodated in the Markov chain framework, though it might be appropriate to increase the number of previous steps in the state description. Regarding stop durations, we could fit a distribution, perhaps Poisson, to each strain's stops. These additions would allow us to attach approximate times to simulated runs.

### **7.3.3: Investigating interactions**

Perhaps the single biggest shortcoming of our conceptualisation of gravitaxis is the way we have considered each fly as an agent operating in isolation. This is clearly not the case, as at the very least other flies in the maze act as obstacles temporarily blocking certain paths. However, interactions are likely to be far more complicated than this, as *Drosophila* exhibit territorial behaviour and male-male aggression. These adversarial exchanges between males are surprisingly complex, involving 'threatening' wing raising, leg 'fencing', holds, retreats and many other behaviours (Chen *et al*, 2002).

A tractable question we can ask is whether flies in the maze show attraction, repulsion or indifference towards each other. The first step would be to identify whenever one fly was in another's 'social zone', i.e. close enough that they can perceive each other, but not close enough for physical interaction. We could then assess whether the trailing fly made the same junction decisions as the leading one significantly more or less often than we would predict if it were alone. Unusually high frequencies of stopping and U-turning on the part of the trailing fly could also be taken as evidence for avoidant behaviour. This analysis would require a substantial modification to the architecture of the annotation program, as it currently just processes traces individually.

If the model was extended to include walking speed (see above), and thus flies' actual positions at any given time could be approximated, interactions could be detected. We could therefore introduce a variable in the state description describing whether another fly was in the vicinity. However, with each addition we have suggested in this and the previous section the state space would grow exponentially, requiring very large amounts of data to estimate probabilities reliably. It may be that a more sophisticated approach than a simple lookup table would be needed, perhaps using some form of clustering to pool probabilities from similar states.

### 7.3.4: Investigating the role of novelty in motivation

We have suggested that wild-type flies may stop more frequently in the latter half of the maze due to the decreasing novelty of the maze environment, based on the observation that memory-deficient mutants fail to exhibit the same pattern. As discussed previously, several alternative explanations would need to be ruled out before we could be confident of this account. If we could demonstrate that increasing novelty towards the end of the maze suppressed this effect, this would strongly support our hypothesis. This could be achieved by changing some feature of the maze halfway through.

Dill & Heisenberg (1995) found that tethered *Drosophila* show a preference for novel visual patterns compared to familiar ones, as measured by the torques they produce attempting to orient themselves towards them. This judgement of novelty seems to be based on a very simple pixel-by-pixel matching process. It is therefore not clear what would constitute novelty in a 3-D environment, but we could try changing the colour, patterning or diameter of the tubing used to construct the maze. It would be important to show that latter-half stopping was reduced regardless of which features were changed (and in which order) to demonstrate that novelty, as opposed to any particular feature of the maze, was causing the effect.

## 7.4: General discussion

Thanks to recent advances such as microarray technology and the sequencing of the *Drosophila* genome, it is becoming increasingly easy to identify genetic bases of complex behaviours such as gravitaxis. The next challenge is to understand the neural machinery underlying the second-to-second execution of these processes. There are many approaches that can be taken to this issue. As we have already mentioned, one can look at spatial expression patterns of genes using techniques such as lacZ histochemical staining. This can help to identify those parts of the central and peripheral nervous systems involved in gravitactic behaviour.

Increasingly advanced techniques for 'genetic dissection' allow us to selectively destroy certain groups of neurons and assess the resultant effects on behaviour. Strauss (2002) describes how this approach has demonstrated the important role the central complex (CX) plays in *Drosophila* locomotor behaviour, and in particular, in co-ordinating the movements of the legs on each side of the fly. Mosaic techniques can be used to confine genetic defects to certain parts of the fly's anatomy. For instance, he shows that flies with an intact brain can compensate for the asymmetry caused by a unilaterally expressed defect in the body, but if the head also carries the defect the fly will walk in circles.

It is difficult to make use of electrophysiological methods to investigate the neural underpinnings of behaviour, as we lack the technology to make *in vivo* recordings from moving flies. However, the ingenious approach of keeping the fly stationary and simulating the environment moving around it has been taken by Lindemann *et al* (2003), who developed a virtual reality flight simulator that measures the attempted movements of a tethered blowfly and generates the appropriate optic flow pattern. This has allowed them to investigate the functional significance of visual interneurons in ways that would previously have been impossible. Another impressively novel technique is that of Lima & Miesenböck (2005), who produce *Drosophila* with optically-gated ion channels in certain groups of neurons, allowing them to be artificially stimulated in freely-moving flies by using lasers.

Modelling is also a vital tool for testing our hypotheses about neural mechanisms. In particular, for behaviours where interaction with the physical environment is important (such as gravitaxis), robotic

models can be especially useful. As they operate as complete systems in the physical world, modellers are unable to study subsystems in isolation or make simplifying assumptions about the input to or output from particular neural circuits (or are at least forced to make these assumptions explicit). For example, Webb *et al* (2003) use a mobile robot to test a relatively detailed neural model of phonotaxis (for the location of mates) in crickets.

If we are to bridge the gap between genetics and complex behaviour, it seems wise to adopt a reductionist approach. Only by studying the various primitive sensorimotor behaviours of a species such as *Drosophila* can we hope to understand how they interact to produce a complex behaviour like gravitaxis or courtship. This study therefore represents a small but important step in this direction, as it has allowed us to move from a monolithic, black-box characterisation of gravitaxis to an analysis of the more basic behaviours that underlie it.



## Appendix: Pseudocode

In this section we give pseudocode descriptions of the tracking and annotation programs. Only complicated functions are given, those that are self-explanatory are omitted.

### A.1: Video tracker

#### *Main program:*

```
bg = getBackground()
juncs = findJunctions(bg)
for i = 1..nflies:
    trace[i]. = new Trace(juncs)           Initialise traces with junction positions
while not allFliesFinished and not endOfFile:
    im = getVideoFrame()
    bitmap = getHits(im)
    blobs = findBiggestBlobs(bitmap, nflies, 5)
    matches = getOptimalAssignment(blobs, traces)
    updateTraces(matches)
saveTracesToFile()
```

#### findJunctions(image):

```
hist = histogram(image)
smoothHist = smooth(hist)
thresh = indexOf(min(smoothHist[55..175]))   Exclude very dark or bright values
for x = 1..xRes:
    for y = 1..yRes:
        if image[x,y] < thresh then bitmap[x,y] = true else bitmap[x,y] = false
spots = findBiggestBlobs(bitmap, 36, 25)
centralSpot = spots.nearestTo(350, 550)
for spot in spots:
    Find those on same row:
    if abs(spot.getY() - centralSpot.getY()) < 15 and spot != centralSpot then rowSpots.add(spot)
neighbourSpot = rowSpots.nearestTo(centralSpot)
scale = abs(neighbourSpot.getX() - centralSpot.getX())
for spot in spots:
    tJuncs.add(new Junction(spot.getX() - 0.303*scale, spots.getY()))
    yJuncs.add(new Junction(spot.getX() + 0.333*scale, spots.getY()))
This does create spurious Y-junctions off the right edge of the maze, but this doesn't matter as no trace can reach their vicinity.
```

#### findBiggestBlobs(bitmap, nBlobs, minSize):

```
for x = 1..xRes:
    for y = 1..yRes:
        if bitmap[x][y] == false then continue
```

```

    thisBlob = growBlob(bitmap,x,y)
    bitmap.setAllFalse(thisBlob)
    if thisBlob.numberOfPixels() >= minSize then:
        blobs.add(thisBlob)
        if blobs.size() > nBlobs then blobs.removeSmallest()
if tracking then: Don't split blobs when finding junctions
    blobs.splitBlobsBiggerThan(42)
    while blobs.size() > nBlobs:
        blobs.removeSmallest()
return blobs

```

```

growBlob(bitmap,x,y)
    blob = new Blob()
    blob.add((x,y))
    checked = []
    toCheck = [neighbours((x,y))] Returns the 4 adjacent pixels: (x-1,y), (x+1,y), (x, y-1), (x,y+1)
    while not toCheck.isEmpty():
        thisPixel = toCheck[0]
        toCheck.remove(thisPixel)
        checked.add(thisPixel)
        if bitmap[thisPixel] == true then:
            blob.add(thisPixel)
            neighbours = neighbours(thisPixel)
            for newPixel in neighbours:
                if (not toCheck.contains(newPixel)) and (not checked.contains(newPixel)) then:
                    toCheck.add(newPixel)
    return blob

```

```

getHits(image):
    for x = 1..xRes:
        for y = 1..yRes:
            bitmap[x,y] = false
            if outOfBounds(x,y) then continue
            if background[x,y] - image[x,y] > 12 then bitmap[x,y] = true
    return bitmap

```

```

getOptimalAssignment(blobs,traces):
    global: bestAssignment = null, bestCost = 10000 Initialise arbitrarily high
    potentialMatchMap = setUp(blobs, traces)
    recursiveMatch(potentialMatchMap)
    return bestAssignment

```

```

setUp(blobs, traces)

```

```

potentialMatchMap = new PotentialMatchMap()
for trace in traces:
    if not trace.isActive() then continue           If fly has left the maze, skip it
    potentials = blobs.getAllInRange(trace)
    if potentials.isEmpty() then:
        potentialMatchMap.makeMatch(trace, null)    Must match null
    else:
        while (potentials.size() > 3):
            potentials.removeFurthestFrom(trace)    Heuristic: can only match closest 3
        potentialMatchMap.put(trace, [potentials, null]) Can also match null
return potentialMatchMap

```

```

recursiveMatch(potentialMatchMap):
    if potentialMatchMap.completed() then:           All traces matched to blobs
        if potentialMatchMap.getCost() < bestCost then:
            bestAssignment = potentialMatchMap.getAssignment()
            bestCost = potentialMatchMap.getCost()
        return
    trace = potentialMatchMap.getTraceWithLeastPotentials()
    potentials = potentialMatchMap.get(trace)
    if (potentials.size() == 1) then                 Can only match null
        potentialMatchMap.makeMatch(trace, potentials[0])
        recursiveMatch(potentialMatchMap)
    else:
        for blob in potentials:
            hypothesisedMatchMap = potentialMatchMap.copy()
            hypothesisedMatchMap.makeMatch(trace, blob)
            recursiveMatch(hypothesisedMatchMap)

```

```

PotentialMatchMap.makeMatch(trace, blob):
    this.assignment.put(trace, blob)
    if blob == null then:
        cost += 1000                                Arbitrary large penalty
    else:
        cost += distance(trace, blob)
    if cost > bestCost then return                 Abort search
    traces.remove(trace)
    if blob != null then:                          Each blob can only match one trace
        for trace in traces:
            potentials = this.get(trace)
            if potentials.contains(blob) then potentials.remove(blob)

```

## A2. Trace annotator

### **Main program:**

velocityWindow = ceiling(1.5\*framerate) *Number of frames in 1.5 sec*

interpWindow = floor(velocityWindow/2)

**for** trace **in** traces:

    trace = interpolate(trace)

    analysis = analyse(trace)

    analysis.removeEventsLeftOfFirstSpot(analysis)

    startTime = analysis.getFirstJunctionEvent().getTime()

**if** spatialAnalysis **then** analysis = removeEventsOutsideZone()

    analysis.combineConsecutiveFalls()

    analysis.combineStopsLessThan1SecApart()

    analysis.save()

interpolate(trace):

**for** i = 0..trace.size():

**if** trace[i] != **null** **then continue**

        j = i+1

**while** trace[j] == **null** **and** j<trace.size():

            j++

**if** j-i <= interpWindow **then**:

            vector = trace[j] - trace[i]

**for** a = 1..j-1:

                trace[i+a] = trace[i] + vector\*(a/(j-i+1)) *Linear interpolation*

**return** trace

analyse(trace):

    scale = trace.getScale() *The distance from a spot to the T-junction to its left*

    curVel = **null** *Last velocity measurement, initialise null*

    fallBlock = 0 *When >0, suppresses velocity measurements*

**for** i = 1..trace.size():

**if** fallBlock > 0 **then** fallBlock--

**if** trace[i] != **null** **and** trace[i-1] != **null** **then**:

*Fall events:*

            vector = trace[i] - trace[i-1]

**if** vector.magnitude() > scale\*1.3/framerate **and** vector.angleTo(down) <  $\pi/4$  **then**:

**if** lastFall != **null** **then**:

*If two falls in quick succession, record the first one as a fall-down*

                    lastFall.setRecovery(down)

                lastFall = **new** FallEvent()

                analysis.add(lastFall)

                fallBlock = velocityWindow+1 *Prevent velocity calculations if window contains a fall*

*Junction events:*

```
if nearJunction(trace[i]) and (not nearJunction(trace[i-1])) then:   Enters junction zone  
    lastEntryPoint = trace[i]  
if (not nearJunction(trace[i])) and nearJunction(trace[i-1]) then:   Leaves junction zone  
    if distance(trace[i], lastEntryPoint) > scale*0.15 then:         Exit different to entrance  
        junction = getNearestJunction(trace[i])  
        direction = getLeavingDirection(junction.getType(), trace[i], lastEntryPoint)  
        analysis.add(new JunctionEvent(junction.getType(), direction))
```

```
if not trace[(i-velocityWindow)..i].contains(null) then:  
    vel = trace[i] - trace[i-velocityWindow]
```

*Stop events:*

```
if vel.magnitude() < scale*0.005*velocityWindow then:           Fly is stopped  
    stopDuration += 1/framerate  
else:                                                             Detect stops once fly starts moving again  
    if stopDuration >= 3 then:  
        analysis.add(new StopEvent(stopped))  
        stopDuration = 0
```

*U-turn events:*

```
if fallBlock == 0 and vel.magnitude() > scale*0.0075*velocityWindow then:  
    Fly moving enough for measurement to be reliable  
    lastVel = curVel  
    curVel = vel  
    if lastVel != null and curVel.angleTo(lastVel) > 3π/4 then:  
        analysis.add(new UturnEvent())  
    if lastFall != null then:  
        lastFall.setRecovery(vel) If first velocity trace after fall, categorise fall accordingly  
        lastFall = null           Then clear the local variable
```

```
else:  
    stopDuration = 0           Clear the stop counter if we have a null trace
```

```
if trace[i] != null then:
```

```
    lastSeen = trace[i]  
    finishTime = i/framerate
```

```
if nearRightEdge(lastSeen) then:
```

```
    analysis.setExitPoint(getNearestExit(lastSeen))  
    analysis.setFinishTime(finishTime)
```

```
return analysis
```



## References

- E. O. Aceves-Pina & W. G. Quinn (1979) Learning in normal and mutant *Drosophila* larvae. *Science*, **206**, 93-6.
- R. R. H. Anholt & T. F. C. Mackay (2004) Quantitative genetic analyses of complex behaviours in *Drosophila*. *Nature Reviews: Genetics*, **5**, 838-49.
- R. H. Anken & H. Rahmann (2001) Gravitational zoology: How animals use and cope with gravity. In G. Horneck & C. Baumstark-Khan (eds), *Astrobiology: The quest for the conditions of life*, Springer, 314-32.
- J. D. Armstrong, M. J. Texada, R. Munjaal, D. A. Baker & K. M. Beckingham (2005) Gravitaxis in *Drosophila melanogaster*: a forward genetic screen. *Genes, Brain and Behavior*, in press.
- J. A. Ashley & F. N. Katz (1994) Competition and position-dependent targeting in the development of the *Drosophila* R7 visual projections. *Development*, **120(6)**, 1537-47.
- T. Balch, Z. Khan & M. Veloso (2001) Automatically tracking and analyzing the behavior of live insect colonies. In *Autonomous Agents*, Montreal, 2001.
- K. M. Beckingham, J. D. Armstrong, M. J. Texada, R. Munjaal & D. A. Baker (2005) *Drosophila melanogaster* – The model organism of choice for the complex biology of multi-cellular organisms. *Gravitational and Space Biology*, **18(2)**, 1-13.
- S. R. Belmain, M. S. J. Simmonds & W. M. Blaney (2000) Behavioral responses of adult deathwatch beetles, *Xestobium rufovillosum* de Geer (Coleoptera: Anobiidae), to light and dark. *Journal of Insect Behavior*, **13(1)**, 15-26.
- S. Chen, A. Yeelin Lee, N. M. Bowens, R. Huber & E. A. Kravitz (2002) Fighting fruit flies: A model system for the study of aggression. *Proc Nat Acad Sci*, **99(8)**, 5664-8.
- M. Dill & M. Heisenberg (1995) Visual pattern memory without shape recognition. *Philos Trans R Soc Lond B Biol Sci*, **349**, 143-52.
- T. Dobzhansky, C. L. Judson & O. Pavlovsky (1974) Behavior in different environments of populations of *Drosophila pseudoobscura* selected for phototaxis and geotaxis. *Proc Nat Acad Sci*, **71(5)**, 1974-6.
- J. S. Duerr & W. G. Quinn (1982) Three *Drosophila* mutations that block associative learning also affect habituation and sensitization. *Proc Natl Acad Sci*, **79**, 3646-50.
- L. Erlenmeyer-Kimling & J. Hirsch (1961) Measurement of the relations between chromosomes and behavior. *Science*, **134**, 1068-9.
- V. J. L. Fourcassie & J. F. A. Traniello (1995) Ant search behaviour analysis with a video frame grabber. *Insectes Sociaux*, **42**, 249-54.
- P. L. Han, L. R. Levin, R. R. Reed & R. L. Davis (1992) Preferential expression of the *Drosophila rutabaga* gene in mushroom bodies, neural centers for learning in insects. *Neuron*, **9**, 619-27.
- J. A. Heward (2005) Flyspotting: Automated insect behaviour analysis. Bachelors thesis, Division of Informatics, University of Edinburgh.
- J. Hirsch (1959) Studies in experimental behavior genetics: II. Individual differences in geotaxis as a function of chromosome variations in synthesized *Drosophila* populations. *Journal of Comparative and Physiological Psychology*, **52**, 304-8.
- J. Hirsch & L. Erlenmeyer-Kimling (1961) Sign of taxis as a property of the genotype. *Science*, **134**, 835-6.
- Z. Khan, T. Balch & F. Dellaert (2004) An MCMC-based particle filter for tracking multiple interacting targets. In *Proc. 8<sup>th</sup> European Conference on Computer Vision*, Prague.
- H. Kuromi & Y. Kidokoro (2000) Tetanic stimulation recruits vesicles from reserve pool via a cAMP-mediated process in *Drosophila* synapses. *Neuron*, **27(1)**, 133-43.

- S. Q. Lima & G. Miesenböck (2005) Remote control of behavior through genetically targeted photostimulation of neurons. *Cell*, **121**(1), 141-52.
- J. P. Lindemann, R. Kern, C. Michaelis, P. Meyer, J. H. van Hateren & M. Egelhaaf (2003) FliMax, a novel stimulus device for panoramic and highspeed presentation of behaviourally generated optic flow. *Vision Research*, **43**(7), 779-91.
- J. Martin (2004) A portrait of locomotor behaviour in *Drosophila* determined by a video-tracking paradigm. *Behavioural Processes*, **67**, 207-19.
- P. A. McMillan & T. R. McGuire (1992) The homeotic gene spineless-aristopedia affects geotaxis in *Drosophila melanogaster*. *Behaviour Genetics*, **22**(5), 557-73.
- L. P. J. J. Noldus, A. J. Spink & R. A. J. Tegelenbosch (2002) Computerised video tracking, movement analysis and behaviour recognition in insects. *Computers and Electronics in Agriculture*, **35**, 201-27.
- J. P. Ricker & J. Hirsch (1985) Evolution of an instinct under long-term divergent selection for geotaxis in domesticated populations of *Drosophila melanogaster*. *Journal of Comparative Psychology*, **99**(4), 380-90.
- J. P. Ricker & J. Hirsch (1988) Reversal of genetic homeostasis in laboratory populations of *Drosophila melanogaster* under long-term selection for geotaxis and estimates of gene correlates: evolution of behavior-genetic systems. *Journal of Comparative Psychology*, **102**(3), 203-14.
- B. M. Spruijt, M. O. S. Buma, P. B. A. van Lochem & J. B. L. Rousseau (1998) Automatic behavior recognition: What do we want to recognize and how do we measure it? In *Proc. Measuring Behavior 98*, Groningen, 264-6.
- S. F. Stoltenberg & J. Hirsch (1996) A gene correlate of geotaxis near *Adh* (2-50.1) in *Drosophila melanogaster*. *Journal of Comparative Psychology*, **110**(3), 252-9.
- S. F. Stoltenberg, J. Hirsch & S. H. Berlocher (1995) Analyzing correlations of three types in selected lines of *Drosophila melanogaster* that have evolved stable extreme geotactic performance. *Journal of Comparative Psychology*, **109**, 85-94.
- R. S. Sutton & A. G. Barto (1998) *Reinforcement learning: An introduction*. MIT Press.
- R. Strauss (2002) The central complex and the genetic dissection of locomotor behaviour. *Current Opinion in Neurobiology*, **12**, 633-8.
- R. Strauss & M. Heisenberg (1993) A higher control center of locomotor behavior in the *Drosophila* brain. *Journal of Neuroscience*, **13**(5), 1852-61.
- B. L. Tempel, N. Bonini, D. R. Dawson & W.G. Quinn (1983) Reward learning in normal and mutant *Drosophila*. *Proc Natl Acad Sci*, **80**, 1482-6.
- D. P. Toma, K. P. White, J. Hirsch & R. J. Greenspan (2002) Identification of genes involved in *Drosophila melanogaster* geotaxis, a complex behavioral trait. *Nature Genetics*, **31**, 349-53.
- R. Voss & J. Zeil (1995) Automatic tracking of complex objects under natural conditions. *Biological Cybernetics*, **73**, 415-23.
- D. Walther, D. R. Edgington & C. Koch (2004) Detection and tracking of objects in underwater video. In *Proc. 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 544-9.
- B. Webb, R. Reeve, R. Horchler & R. Quinn (2003) Testing a model of cricket phonotaxis on an outdoor platform. *Towards Intelligent Mobile Robots (TIMR '03)*, Bristol.
- F. W. Wolf, A. R. Rodan, L. T.-Y. Tsai & U. Heberlein (2002) High-resolution analysis of ethanol-induced locomotor stimulation in *Drosophila*. *Journal of Neuroscience*, **22**(24), 11035-44.