University of Edinburgh
Division of Informatics

Cricket visual homing

Undergraduate Dissertation
Cognitive Science

George Buckenham

April 1, 2009

Abstract: Crickets have previously shown the ability to locate a hidden cool spot in a heated arena [Wessnitzer et al., 2008]. A new paradigm was designed to replicated these results in which light levels were changed as the cricket moved across the arena. No learning effect was found, but the crickets had a significant preference for the quarter of the arena in the lights adjacent to the boundary.
Three visual homing algorithms, Average Landmark Vector [Lambrinos et al., 1998], Warping [Franz et al., 1998] and Gradient Descent [Zeil et al., 2003] were implemented in four sets of images, three of which were captured within the cricket arena. A robotic implementation was also developed. It was found that the best performing was ALV but only with artificial landmarks. The consistently best performer was Gradient Descent, although Warping was not unsuccessful. Finally, the applicability of these findings as algorithms for crickets and robots was discussed.

Acknowledgements

Thanks must go to Barbara Webb for all her kind help throughout, and to Michael Mangan for his generosity and friendliness. And thank you to my proofreaders, notably Dad and Rachael.

Contents

0.1 Introduction

The aim of this project is to investigate possible mechanisms by which a cricket can perform visual homing. To this end, I developed a new paradigm in which to test a cricket's ability to learn the location of a home location. Their performance on this was to be compared against the performance of various algorithms upon a dataset collected from this arena, and a robot platform navigating within the same arena.

The paradigm I designed was an adaption of the Tennessee Williams paradigm, which was first used by Mizunami et al. [1993], and aimed to replicate the Morris water maze paradigm [Morris, 1981] for cockroaches. Both paradigms consist of an aversive condition which is eased in one invisible spot. A creature can be said to successfully perform visual homing if they are capable of navigating to the hidden spot. In the Morris water maze experiment, the aversive condition was water, and the hidden spot was a submerged platform (allowing the rat to cease swimming). In the Tennessee Williams paradigm, the aversive condition is a heated metal floor and the hidden spot is a small cooled area.

Wessnitzer et al. [2008] , following on from suggestive results from Scotto-Lomassese et al. [2003] repeated this design with the crickets *Gryllus bimaculatus*. They found that crickets quickly learnt the location of the cool spot (although they would not always directly seek it). This was due to visual cues: the learning effect was not present when the crickets were deprived of all visual cues. This was confirmed by their response to the backdrop being rotated: they approached the point corresponding to the cool spot had there been no rotation. There were effects corresponding to the visual cues the crickets were given: they performed best when there was a detailed natural scene present, second best when there were no cues but blank white walls, third best when there were artificial landmarks attached to the walls, and worst when there was no light at all.

After a series of pilot experiments, I designed a new paradigm that changed the light conditions depending upon the location of the crickets. When they were in the aversive section of the arena, bright lights were shined upon them. When they were in the less aversive spot, the lights turned off. In order to implement this, I designed a system capable of tracking crickets via an overhead webcam. Operating via background subtraction, this directly controlled a bank of lights arrayed above the arena.

To test this paradigm, I carried out a batch of experiments where the arena was bisected into light and dark halves. These showed that the crickets did not find the light an aversive stimulus, but rather preferred the section of the arena in the light near the boundary. Unfortunately, they also did not show change in the proportion of time spent in any sector. While this does not disprove learning, it also offers no evidence for it. The performance of the algorithms modelled were therefore compared against that of Wessnitzer et al. [2008], who used the same arena.

Most algorithms developed take the Cartwright and Collett [1983] Snapshot Model as a base. This says that, given a current image and a image taken at the home location, it should try to match landmarks on both images to each other. At this point, trigonometry can recover the distance and direction you have moved in from the displacement (and possibly rescaling) of the landmark across a visual field. Assuming correct matching of landmarks, this method will reliably iterate to return the agent to the home location.

The images used as input are omnidirectional panoramic images, typically taken by a camera facing a curved mirror. These images were then unwrapped to produce a fully panoramic image.

This landmark matching has been adapted to work on a number of landmark features. For example, using small image blocks (or even single pixels) as features, using overall displacement of the image via optical flow [Vardy and Mller, 2005], or using Scale Invariant Feature Transform (SIFT) features.

It is also possible to use an average of all the landmarks detected, and compare this across locations. This is known as the Average Landmark Vector model [Lambrinos et al., 1998], and was implemented in §2.1.1

Another technique that is used works upon the observation that there is a smoothly changing gradient of image differences. By moving such that this difference decreases, the agent moves closer to the home location [Zeil et al., 2003]. An estimate of the gradient can be taken by sampling three adjacent and orthogonal points. This was another of the algorithms implemented(§2.1.3).

Instead of sampling adjacent points, the gradient may be estimated using optical flow techniques to calculate the distortions caused by possible movement[Moller et al., 2006]. In a similar fashion, the displacement of the current image that would be caused by an arbitrary journey from the current point can be estimated. By comparing a large range of combinations of rotation, displacement and bearing of travel, the travel which is most similar to the home location may be found. This is known as the Warping method of Franz et al. [1998] (This was implemented in §2.1.2)

These three implemented algorithms were tested against a series of datasets taken in the cricket arena and one publically available dataset [Vardy and Moller, 2005]. They were assessed both by viewing their performance directly and in a series of trials which assessed their performance travelling from one location to another. These were done from every location in the arena to every other location, and metrics of performance were taken: Average Angular Error (between the ideal bearing and the produced bearing) and Rate of Return(the number of trials that successfully returned to the original position).

It was found that when artificial landmarks were present, ALV performed best, closely followed by Gradient Descent. When there were Blank Walls, none of the algorithms performed at all well, but Gradient Descent had the best performance. In the Natural Scene dataset, Gradient Descent performed best.

These algorithms were also implemented on a robotic platform. This appeared to work well, although it was not systematically tested.

1. Cricket Trials

1.1 Introduction

One of my aims was to repeat the Wessnitzer et al. [2008] experiments using a different paradigm. This would provide independent confirmation of their results, and, it was hoped, inform my comparisons of robotic algorithms with their behaviours. Multiple pilot experiments were performed, in order to test possible paradigms (Appendix A). The paradigm that was chosen was that of varying the illumination of the arena in such a way as to correspond to the position of the crickets. The hope is that crickets, having a natural affinity for the dark, would prefer the dark areas of the arena, and thus exhibit homing behaviour. While in the pilot experiment, I varied the illumination by hand, this was not feasible for controlled trials. I also needed to record the position of the cricket, for later analysis of their movements.

1.2 Overview of system

In order to vary the lights with the crickets position, an image of the arena was first captured from an overhead webcam. This was processed to remove extraneous information, and then subtracted from a stored background image to locate the cricket. Depending on the crickets position, lights were activated or deactivated via a computer controlled relay board. This change in illumination also informed the background subtracted from the current image.

1.3 Details of Equipment

The arena I used was the same used in Wessnitzer et al. [2008]. This arena consists of a circle of 40cm in diameter, placed on a thin sheet of metal. This was placed upon a heated tank in the original experiments. This metal sheet was marked out with 1cm subdivisions due to earlier capture of panoramic images of the arena by Michael Mangan. These markings proved impossible to remove. In order to remove these markings I inserted a thin sheet of paper. This I judged preferable to having visible markings remain, but, being more absorbent of odours than metal may have let the crickets leave scent trails. Above the arena was a wooden frame 92cm in height, with a bar running diagonally across the center from which a Logitech webcam is mounted. Hanging from this upper platform is a circular curtain which conceals the room from the floor of the arena. To further insure isolation, blackout curtains are attached around the outside of the frame. Lights were arrayed around the platform facing down. As work progressed, more and more lights were added - initially only a few desklamps were used, but by the end a fluorescent bulb, a halogen floodlight and 3 desklamps were arranged around the outside, as well as a infra-red light (discussed later)

1.3.1 Controlling the lights

The output of the motion tracking is switching of lights. Due to the high power demands of the lights, it is not possible to drive them directly from the computer. Instead, a relay board was needed. After consultation, I purchased a parallel port controlled relay board. This received a 5V control signal from a computers parallel port and used it to switch a relay which could handle mains voltage . Originally the intention was to us a university supplied DICE machine with a parallel port to send this control signal. However, sending this arbitrary signal requires intervention from a user with “root” access, which could not be obtained. It was therefore necessary to send the 5V control signal via a “Phidgets” interface board [1], which was on hand. This was in turn controlled by a USB connection and the available “Phidgets” library. Using this library, a small C utility was written, set_pin1 , which took as an argument either “1” or “0” for “on” or “off”, and which was controlled from MATLAB.

The relay board was used with a normal household extension cord. This provided 4 simultaneously switchable sockets. Using this extension cord instead of directly splicing into a lamp’s cord enabled me to vary the quality and type of lights used. It also had a LED indicating whether power was being supplied to it or not - something that came in handy with sometimes unreliable lamps.

1.3.2 Image capture

To capture images of the cricket I used a standard USB webcam. This was usefully already mounted on a frame above the arena. This I retrieved frames from using the mplayer utility , using the following command:

```
mplayer tv:// -tv driver=v4l:width=320:height=240:device=/dev/video0
-saturation -100 -vf framestep=4,crop=190:190:60:10 -vo pnm
```

This, decomposed, takes a stream from `/dev/video0` using the `v4l` library (`mplayer tv:// -tv driver=v4l:`) at 320*240. It also reduces it to a grayscale

---

[1] PhidgetInterfaceKit 8/8/8 (http://www.phidgets.com/products.php?category=1&product_id=1018)

image, something done here because for the speed benefits as opposed to doing this in MATLAB. (Note that this unfortunately does not change the type of the image to grayscale - however, all three channels are the same, so discarding all but one channel produces the same result.) It also applies (`-vf framestep=4,crop=190:190:60:10`) video filters in order to crop the image to a square centered on the arena itself - this was done to reduce the filesize, and (in earlier versions of the code) prevent movement in the background from interfering with the cricket detection. It also reduces the framerate to one in four frames received from the camera - again this is used to reduce the eventual size of the images stored, and allow MATLAB sufficient time to process each frame (although it copes gracefully with receiving more frames than it can process). This could easily be increased should the framerate prove to be a problem, but this framerate seemed to be quite sufficient. Finally, `-vo pnm` outputs the resulting stream into a series of consecutively numbered portable pixmap images (ppm). These are stored in the directory the command is run.

To transfer this image data into MATLAB I wrote a function, `latest_im.m`, that takes a sequential number and a directory, and retrieves the image corresponding to that offset from the directory. This image is reduced to a single channel, and then passed back to the main function for processing. The sequential number is obtained by taking an offset from the total number of files listed in the directory - if it is greater than last time the loop was checked, a new image is retrieved. In this way, the latest stored frame is captured.

1.3.3 Thresholding and motion detection

To detect the cricket position, the following algorithm is used (implemented in `crick_loc.m`):

```
Inputs:
curr_image : 190 by 190 array of brightness values
bg_image : 190 by 190 array of brightness values
mask: 190 by 190 binary mask array

for each frame:
        curr_image = gaussian_smoothing(curr_im)
        bg_image = gaussian_smoothing(bg_im)
        image = curr_image - bg_image
        image = image * mask
        current_position = index_of_max_point(image)
```

Given the constraints placed upon the input images, this simple system is sufficient. The Gaussian blurring is applied because the camera feed is very noisy, especially under low light conditions. This noise can occasionally be greater than the difference between the dark of the cricket and the light of the arena floor. The cricket, however, is the only region with a consistent difference. The mask is applied due to earl difficulties - since the image supplied is square, and the arena is round, even with the tightest possible crop, the corners of the image showed the curtains surrounding the arena. If these curtains were moved between the background image and the current image, then they would register as a difference, and usually end up being consistently favoured over the cricket (being a relatively larger target).

This simple subtraction algorithm is fed by `run_detector.m`. This loads the background images and the binary crop mask and then enters an eternal while loop. While in the loop, the latest frame is grabbed, and `crick_loc.m` is ran on it. The background mask is chosen to correspond to the internal representation of the light level. If the crickets location has changed from one region to another, the light is changed by calling the `set_pin1` utility. Finally, the current data - cricket co-ordinates, current state of the light, and timestamp is appended to a matrix, and the matrix is then saved to disk. The current image is displayed, with the different regions displayed and the current estimated cricket position marked.

Before each trial was run, it is necessary to capture a background image under both light conditions, set the target area and manually mark the region corresponding to the arena floor. For the sake of sanity, this was scripted. This was also stored individually each time I ran a trial, in order to reproduce the image subtraction later. These masks had to be collected each time as small shifts (in the camera position, for example) led to edges being seen as differences. The camera mount, while stable enough for each run was not secure enough to avoid cumulative error from being introduced: the data collected would become worse upon each trial unless new masks were regularly collected. The crop mask collected here is applied on top of the crop performed by mplayer.

One recurring problem was that of blooming. When the light levels change from high to low, the camera takes some time to change exposure levels. This leads to a few frames of being completely oversaturated and containing only white pixels, or being undersaturated, and “streaky”. Obviously this is a problem for the background detection, as the cricket is not visible. Consequently the location of the cricket is estimated as being the location in the background image with the greatest difference from the white or black values currently being read as the current image. This problem can be particularly bad if it causes the new cricket location to be estimated as in the other region. This can lead to a loop where the lights quickly alternate, only locating the correct position of the cricket (in the dark area) when the lights are on. This "strobing" is not desirable as it muddies the association between location and light level for the cricket.

To solve this, I introduced thresholds. For each frame, the mean pixel brightness was calculated. If this fell within 10 steps of the brightness of a typical frame in this light condition then the frame was used. If not, it was ignored and the cricket was assumed to have maintained its previous position. Typically, after a few frames the camera would have adjusted to the new light level and tracking could continue. Values were not taken from the background images because of issues both from noise and from issues where the brightness of the illuminated image seemed to oscillate slightly. This oscillation was presumed to be an artifact of the autoexposure function of the webcam I was using.

This helped resolve the problem, but was less than ideal. The system lost all sensitivity to the crickets movements at just the time that they were near the dividing line. The camera used had previously been modified to remove the infra-red filter common in webcams. This meant I was able to add a visible light filter to the front, and have its sensitivity largely restricted to the infra-red range. With the addition of a constant infra-red light source (a normal lamp with an infra-red bulb added) also filtered to remove visible light, the illumination experienced by the camera is far more constant. Since crickets are not sensitive to infra-red light, this illumination does not affect them. This resulting system still has variations in illumination, but they are reduced. This causes a further increase in performance very cheaply, and allows the dark condition to be closer to complete darkness.

1.3.4 Limitations

The system designed serves its purpose: It tracks a cricket, modifies light conditions, and continues tracking the cricket in these modified light conditions. Due to the nature of the task, it can occasionally fail: sometimes it detects stray matches: even worse, the stray matches can be consistent, leading to the "strobing" loop described above. In my final use of the system, 3 out of 55 runs suffered from this problem. I believe this is largely caused by not taking enough care resetting the masks betwen each run.

This does not count stray matches which did not interfere with the switching. I estimate that 225 of 34760, or 0.65% frames suffered from stray matches with less noticebale results. This estimate was obtained by looking at the crickets velocity - if the cricket was moving beyond a certain threshold, then it was assumed that the match was errant. This assumption will not hold in all cases. This same system of estimation could have been used to remove these stray matches online - the task is complicated there by the necessity to have a location at all times for the cricket - if the cricket does travel faster than the threshold, the system will have trouble fixing onto it again. However, this trick is unlikely to work in resolving the persistent “strobing” errors - a persistent error will last longer than a frame and so will be seen as stationary and hence valid.

The system is not very readily generalised beyond the specific task it is designed for: it can only track one object, it requires there to be contrast between the ground and the target, it is not tolerant to shifts in the camera. It is also not especially accurate: it only finds one spot within the target, does not find a consistent point, for instance the centroid. As a result, a stationary cricket may be interpreted as a cricket that is shifting slightly. However, for tracking single objects against a stable background from a stationary position it works effectively - this added precision is not needed for the purposes of later trials.

Another desirable feature would be an ability to track the direction the cricket is facing - this is readily visible with human eyes from the image feed, and in any case the cricket’s movement could be used to provide confirmation. If the cricket rotates, this would strengthen the case for any algorithm that requires rotation (but a lack of evidence not weaken it: the cricket may merely do this mentally).

1.4 Trials

While cricket behaviour was observed while the setup was developed, it is impossible to gain conclusive results from casual observation. This is due to individual differences between the crickets, inconsistent interpretations of their movement, and a lack of consistent application of stimuli. To properly evaluate their movementit is necessary to view multiple crickets under controlled conditions. To this end, a series of trials was ran.

The arena was bisected into a “light” arena and a “dark” area of equal sizes. From this it is easy to compare preferences for each state of the light. Similarly, it is easy to compare the time spent close to the boundary area. It is only slightly more difficult to compare the velocity of the cricket at every position in the arena. On the final trial, the landmarks were rotated by 90 degrees. If the crickets retain a knowledge of their position relative to these landmarks they should now have changed preferences. This final condition is not necessarily very reliable: Wessnitzer et al. [2008] showed that crickets could use subtle light gradients in the screening curtain above the landmarks. Due to the addition of a non-uniform source of light above the arena, this problem has if anything increased. In addition, the crickets will be acquiring information about the areas which produce light and dark from their movements in the final trial.

The experimental hypotheses that will be tested are the following:

- The crickets have a preference for the dark
- The crickets have an aversion to the boundary and the sudden change of light
- When the crickets are in the light half of the arena they will prefer to remain close to the boundary
- The crickets will display learning behaviours - improving their performance in later runs
- The crickets will maintain their knowledge about position of the areas of light and dark relative to the rotated background

1.4.1 Procedure

The crickets used were fully-developed female *Gryllus bimaculatus*. These were fed dog-food and water.

Each cricket was placed in the center of the arena and allowed to roam freely for 5 minutes, whereupon they were retrieved and rested in a cup for 2 minutes. This was repeated ten times, plus one additional trial with the background rotated. The background was a natural landscape featuring a tree and a rock formation (The same as used in Wessnitzer et al. [2008]). Only one background was used: the landscape was chosen as it was the one the cricket were most successful at matching against in Wessnitzer et al. [2008]. Originally, this was to be repeated for ten crickets, however it was only performed on five. This was due to a disease affecting cricket breeders resulting in the unavailability of adult *Gryllus bimaculatus*.

1.4.2 Results

1.4.2.1 Descriptive

As a result of these trials I now had location data for every frame captured over 55 trials (11 trials for 5 crickets). I also had the light state, the time offset for each frame (frames were not always captured with perfect regularity), and the original captured frame. From these, I produced this data for every trial: the number of frames in the light half of the arena, the number of frames in the dark half of the arena, the number of frames in the middle (close to the boundary) and the number of frames not close to the boundary. I also produced the average velocity moving from each frame. To eliminate the stray fixations I excluded those frames with a velocity above a certain threshold.

Those trials where the detector did not work properly were also excluded. Luckily, of the three times where the lights suffered from persistent “strobing”, two were on their tenth run (Crickets 2 and 3) . Cricket 1 was subjected to strobing on his fifth run: this datapoint is not present, but subsequent runs are. This seemed justifiable, as no identifiable change could be seen in its behaviour in subsequent runs.

From the captured data a heatmap of frequency of positions can be generated (Figure 1.4.2.1). This shows that, in accordance with expectations, the crickets preferred to run around the outside of the arena - the wallfollowing behaviour observed in Wessnitzer et al. [2008]. There is also a cluster towards the center - this is at least partly attributable to the crickets being placed there at the beginning of each run (they then often wait there for some time). However it is not entirely attributable - the crickets did diverge from the walls of the arena in that location. If the initial placement of the crickets were the only factor, there would not be the small scattering of points between the walls and the center in the light half near the boundary. This can be confirmed by viewing the actual footage of the runs.

The locations where the crickets spent the most frames are next to the walls, and the boundary, in the “light” side. This may well be attributed to a sudden freezing reaction of the crickets to the change in light intensity. This can be graphically seen in Figure 1.4.2.1 - a plot of average velocity throughout the arena. The average velocity around the wall is lowest at that point. This does not entirely explain the reaction, as the cricket are observed to move slightly, and then continue remaining in the same position. It is interesting to note that this does not occur (at least not nearly to the same extent) when they run into the “dark”.

Figure 1.4.2.1 also shows that the crickets move faster when they are in the dark. It also seems to show a reduction in average speed at the furthest reaches from the boundary - these are not matched by corresponding increases in frames spent there, so the average is not decreased by the crickets resting there (again, the footage appears to confirm this).

While there was some variation betwen crickets, they all showed the same preferences for areas within the arena. This suggests that the preferences shown are valid and would persist in a larger trial.

1.4.2.2 Hypothesis testing

- The crickets have a preference for the dark

The crickets showed a preference for the light over the dark half of the arena. In

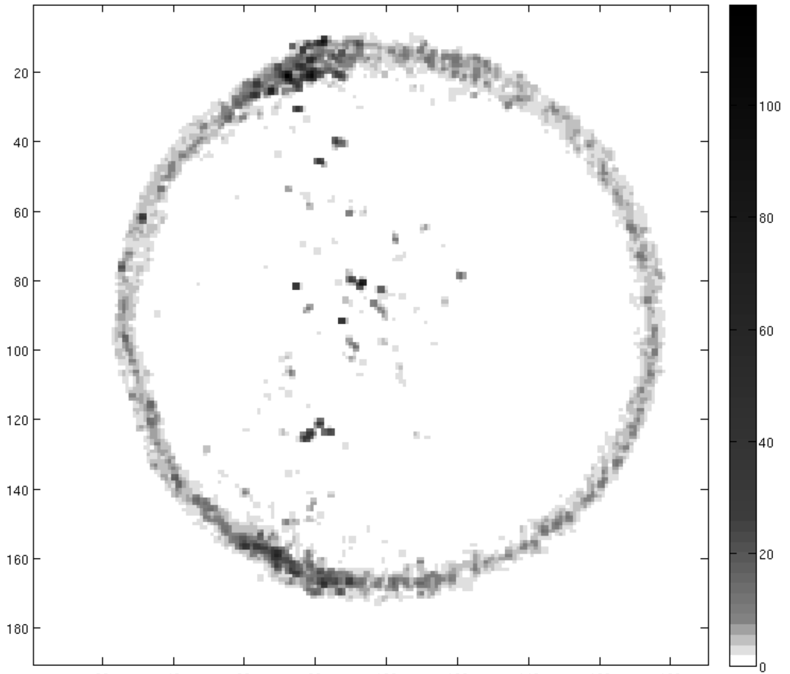

Figure 1.1: A heatmap showing the cumulative position of the crickets over all frames (Blurred slightly for clarity)

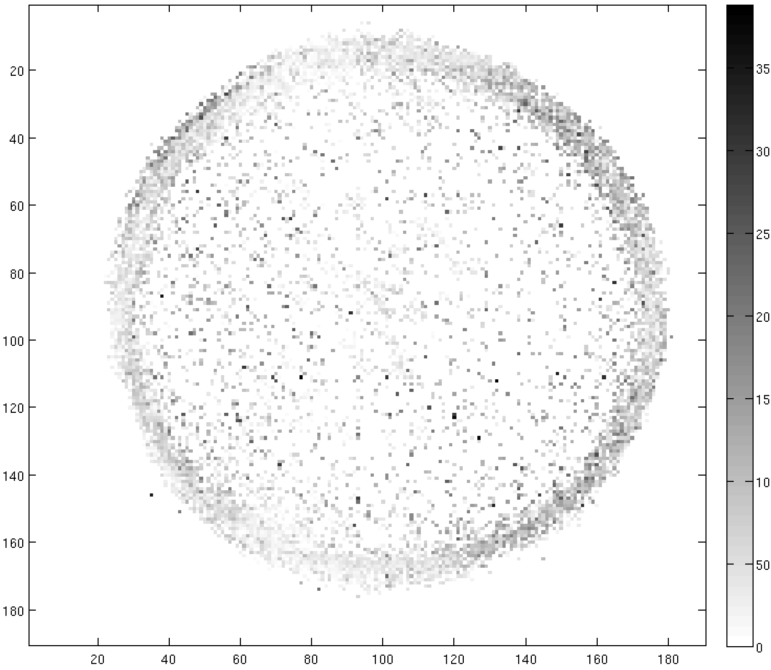

Figure 1.2: Heatmap of average velocity at each point in the arena

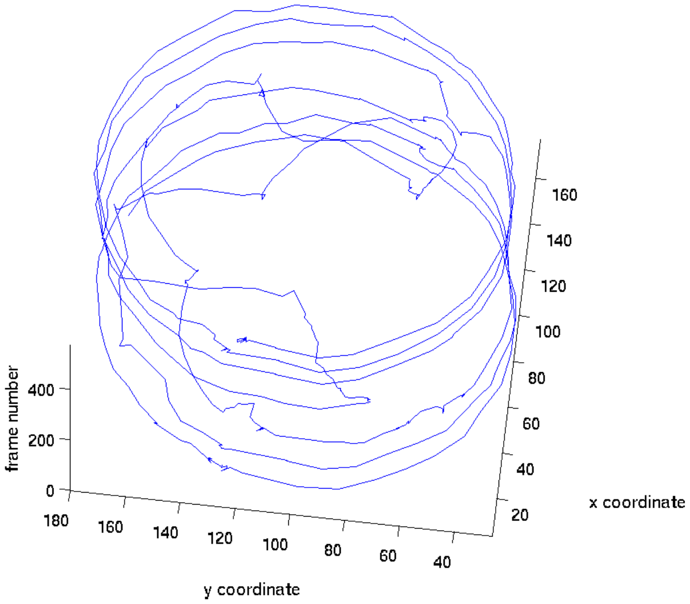

Figure 1.3: A typical cricket path in a single run (Cricket: 4, Run: 4)

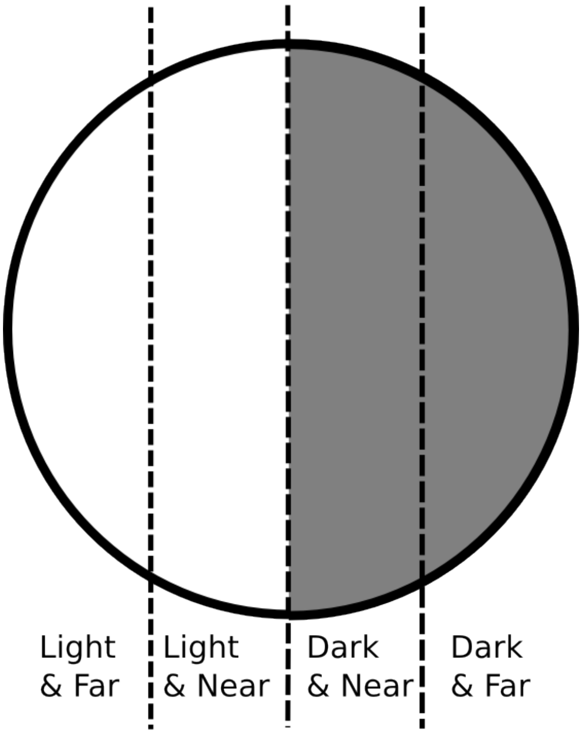

Figure 1.4: A diagram showing the 4 sectors. Note that each is of equal area.

total 60.91% of frames captured had them in the light half of the arena. This disproves the first hypothesis.

This can be shown from a Wilcoxon Signed Ranks Test over all runs, which showed a significant difference (p ¡ 0.0001).

- The crickets have an aversion to the boundary and the sudden change of light

58.64% of frames captured had them in the half of the arena nearer to the boundary. This measure is likely to be a at least slight underestimate, as the capturing software loses track of the cricket for a few frames after they change from one half to another. However, looking at the preference for the light and dark near boundary sectors individually, one can see that there is a definite preference for the light half, and an aversion for the dark half. Crickets spending an increased amount of time in an area does not necessarily indicate a affinity for that area - it may instead indicate a startle response of freezing to the spot, which is also suggested by the raw footage. This is also borne out by the velocity data (shown in Table 1.4.2.2) - The average speed near the boundary in the light half of the arena is the lowest of all four sectors.

This can be shown from a Wilcoxon Signed Ranks Test over all runs, which showed a significant difference (p ¡ 0.0001).

- When the crickets are in the light half of the arena they will prefer to remain close to the boundary

This hypothesis does appear to be true. In 41.72% of the frames captured, crickets were in the quarter of the arena which was in the light and adjacent to the boundary. As noted above, this may be due to a freezing behaviour as a response to the changing light conditions. However, it does not appear that this is the only factor that contributes to this.

This can be shown by a Friedman non-parametric (ranked) ANOVA over all crickets, which showed a significant difference. (p ¡ 0.05)

- The crickets will display learning behaviours - improving their performance in later runs.

As Table 1.4.2.2 and Figure 1.5 show, there is no noticeable learning effect over the course of the ten trials. This does not mean the crickets are not displaying learnt behaviours: only that their learning does not operate on the same scale as the ten trials. Responses may be learnt within a few exposures to a change of light conditions, and thus halfway through the first trial, which would not be

Light & Far	Light & Near	Dark & Near	Dark & Far
19.19 %	41.72 %	16.92 %	22.17%

Table 1.1: Proportion of frames all crickets spent in each of four equal sections of the arena: near boundary or further from boundary, in "light" side" or "dark" side"

Cricket no.	Light & Far	Light & Near	Dark & Near	Dark & Far
1	16.74 %	48.42 %	12.76 %	22.08 %
2	12.76 %	52.52 %	15.66 %	19.06 %
3	31.88 %	31.89 %	18.19 %	18.04 %
4	14.67 %	39.19 %	19.14 %	27.00 %
5	20.10 %	38.36 %	17.88 %	23.66 %

Table 1.2: Proportion of frames each cricket spent in each of four equal sections of the arena: near boundary or further from boundary, in "light side" or "dark side"

apparent from these metrics. [2] These metrics are not particularly sophisticated: the crickets show no signs of finding any of the sectors extremely aversive, and hence may have knowledge of the location of the boundary line without specifically avoiding any areas.

- The crickets will maintain their knowledge about position of the areas of light and dark relative to the rotated background

This knowledge would presumably be seen as a variation in the pattern of distribution. This is not seen in the proportion of frames spend in each sector of the arena, which resembles closely those of the unrotated arena. Neither can a differing pattern be seen in the heatmap of where frames were spent in the arena.

When compared against the time spent in each sector:

Entire arena	Light & Far	Light & Near	Dark & Near	Dark & Far
41.4684	50.3519	17.7124	37.6295	81.4173

Table 1.3: Average cricket velocity for each sector, and over the entire arena

---

[2] In Wessnitzer et al. [2008] not much learning effects were found after the first trial, so this is not unlikely.

Light & Far	Light & Near	Dark & Near	Dark & Far
0.1463	0.4869	-0.2793	-0.7057

Table 1.4: Correlation coefficient between velocity within a sector and frames spent within it by each trial

Run no.	Light & Far	Light & Near	Dark & Near	Dark & Far
1	17.71 %	39.94 %	19.98 %	22.38%
2	19.83 %	40.63 %	16.91 %	22.63%
3	23.56 %	37.33 %	15.68 %	23.44%
4	19.97 %	38.87 %	14.91 %	26.25%
5	20.60 %	43.26 %	16.61 %	19.53%
6	18.89 %	39.58 %	21.11 %	20.42%
7	13.75 %	48.31 %	20.35 %	17.59%
8	21.51 %	45.25 %	11.41 %	21.83%
9	19.65 %	43.80 %	14.72 %	21.83%
10	15.65 %	38.44 %	17.08 %	28.82%

Table 1.5: Proportion of frames all crickets spent in each of four equal sections of the arena in each trial: near boundary or further from boundary, in "light side" or "dark side"

Light & Far	Light & Near	Dark & Near	Dark & Far
25.49%	40.36%	16.64%	17.52%

Table 1.6: Proportion of frames all crickets spent in each of four equal sections of the arena for trials with rotated arena: near boundary or further from boundary, in “light” side” or “dark” side”

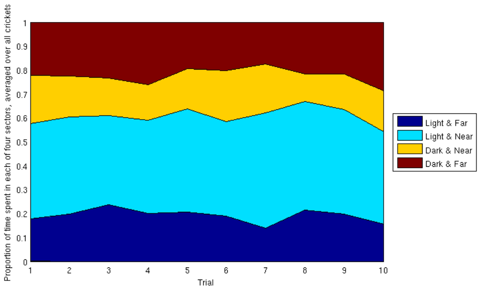

Figure 1.5: Proportion of frames spent in each sector over ten trials. Figure given is averaged over all crickets

1.4.3 Discussion

It was disappointing to fail to find any learning effect over the trials. It is, as said previously, still possible that learning was performed. One means by which this could be tested would be having a final trial in which no rotation was performed and the light switching was not running. This would test whether there was any residual preference for the sectors, without overwriting these preferences with new input from the light changing.

It would also be informative to test the crickets when the lights were constant, both in the light and dark. This would provide a baseline of activity to compare against: i.e. if the crickets have reduced velocity when it is dark, then this provides an explanation of their reduced velocity in the dark half of the arena. If a similar pattern of preferring to spend time in the “Light and Near” sector is found despite the light not varying, then it makes sense to attribute this pattern to the background and not to the lights. However, since the rotated trials showed the same pattern, this does not seem very likely.

The finding that crickets did not prefer the “dark” half of the arena was a surprise: crickets appear to prefer dark places and small holes where they may hide. [Walker and Masaki, 1989]. This can even be shown in their preferential heading towards a dark landmark [Wessnitzer et al., 2008, Honegger and Campan, 1989]. That this response is shown to be heightened by relatively short black landmarks implies that this is used to seek out suitable hiding places. This scototaxic (dark-seeking) behaviour may not truly be a preference for dark, but only a instinctive movement towards dark things. This would explain their lack of attraction towards the dark half of the arena. Alternatively, the crickets may indeed prefer dark crevices even when they are not visible until they have been entered. However, this half of the arena was only dark, it was not enclosed. It is possible that the crickets were able to sense this (perhaps by some small light traces from the canopy, perhaps by some other sense), and hence did not find themselves attracted to the “dark” side.

The strongest, and most striking finding is the crickets preference for the sector of the arena near but not in the “dark” side. It cannot be entirely ruled out that this is due to freezing behaviours brought on by the sudden presence of light. These appear similar to the thanatosis described in Nishino and Sakai [1996], Nishino [2004]. It does not appear as long lasting, however, and is followed by occasional small movements which are entirely uncharacteristic of thanatosis. That there is a real preference for this area can be seen in the seeming preference for a path along the middle of the arena in the light. I would hypothesise that this is due to the cricket’s desire not to stray too far from the safety and concealment of the dark, which at the same time continuing with the exploration of the arena. Such a hypothesis could be tested by applying a threatening stimulus to the cricket and observing its response. Such a threatening stimulus would be difficult to generate: most threats can be localised to a direction by a cricket by means of its sensitive vibration detectors. The abilities of these are documented in Dangles et al. [2007] (albeit in another species), where they have been shown to detect the direction of approaching predators at some distance. Producing a convincing threat that does not produce this directional information would be challenging (perhaps from directly above?), as well as indicing the previously discussed thanatosis.

The crickets again showed dogged persistence in pursuing a strategy of thigmotaxis, or wallfollowing. This has been observered in many previous arena set-ups involving crickets [Wessnitzer et al., 2008, Kastberger, 1982]. This time it appeared moderated by the differing conditions between the two sides of the arena. Not only was their speed interrupted when they crossed the boundary line, they also showed an aversion to the two ends of the arena furthest from the boundary. (as can be seen in Figure 1.4.2.1). That this data can be collected from crickets which are locked into a pattern of thigmotaxis is valuable - this time is not excluded from the data.

It can be seen that in several ways the results obtained appear to differ from those obtained in the pilot experiments. While this may be partially attributed to more rigorous experimental methods, there are other possible causes which suggest directions for future work. Notably the reaction of the cricket towards the dark area appeared to change. This may also be attributable to the varying intensities of light produced as the setup matured. Initially there was a strong bright light and moderate dimness. When the light was moved further away the intensity of the light decreased according to the power-law, and hence produced much less contrast. This was noticeable in a lessening of the crickets reactions to the varying light. In order to increase the contrast, the dark condition was made darker (by use of blackout curtains) and the light condition was made lighter (by the increasing addition of yet more powerful bulbs and lamps). The end result was an even more marked contrast between light and dark than existed in the initial pilot experiment. It is quite possible that the crickets responses to complete, or near-complete dark differ from the crickets response to dimmed light. Varying the different intensities of the light and the dark conditions may produce different results.

In the initial pilot experiment, a small area was switched to dark as opposed to an entire half of the arena. This was my eventual intention with the more developed setup - however the time constraints of perfecting the cricket detection systems and running the experiments themselves prevented this. Evaluating the crickets performance in this system it can be seen whether there is a definite attraction to the area near the dark - the crickets have no cause to travel into the centre of the arena (and will not, usually, being transfixed with thigmotaxis). If a cricket does, it does not tend to pause there for any length of time - preferring corners, if not crevices. If this setup was tried and these behaviours were found, this would provide even stronger evidence that crickets have a preference for remaining near dark areas.

One aspect of my experimental setup I was not happy with was the use of a paper covering for the floor. This readily allows scent traces, as well as signs of wear caused by the crickets themselves. A smooth, wipeable surface is preferable. I do not, however, think this was a large factor in my results - the response of crickets between the first and last trial were fairly consistent and did not show an aversion or preference learnt between crickets for any areas.

2. Modelling & Algorithms

In this section, I evaluate the plausibility of several algorithms of visual homing as models of cricket visual homing. I have also attempted to determine the suitability of using such algorithms for homing on a robotic platform. To do this, three algorithms have been implemented and tested against 3 datasets recorded in the cricket arena and one commonly used publicly available one. These have also been implemented on a robot platform, which attempts to home within the cricket arena. The three algorithms I have implemented are the Average Landmark Vector (ALV) model [Lambrinos et al., 1998], Warping [Franz et al., 1998], and Descent in Image Distances [Zeil et al., 2003].

2.1 Algorithms

2.1.1 Average Landmark Vector (ALV)

The Average Landmark Vector (ALV) model [Lambrinos et al., 1998] is a simplification of the Landmark model of Cartwright and Collett [1983]. While the Landmark model attempts to determine a vector based upon all the landmarks present in a scene, the Avergae Landmark Vector averages these by finding the average of all of these to produce a single virtual landmark.The homing bearing can be found by subtracting the vector from the home location to the landmark from the vector from the current location to the landmark. his is an improvement upon the Landmark Model because it requires less storage of information: only a single vector must be stored to find a route back to the home position. It also negates the matching problem: it is not necessary to identify landmarks, only to separate them from the background.

This does trade off with the requirement for a reliable compass direction to be known for ALV to work. This is necessary to work out the relative positions of landmarks from the home position and from the current position. This is not a huge ecological obstacle for the cricket and the desert ant in the wild (which the algorithm was designed to match) since they both have polarised vision and can thus extract compass information from the sky [Homberg, 2004]. However, in Wessnitzer et al. [2008] the crickets were capable of homing without a source of polarised light, so this cannot be the entire explanation. A “visual compass” can be used instead, which works in much the same way as the gradient descent method(§2.5.2). This is unreliable and carries a large overhead (it is necessary to keep a detailed representation of the home position, frustrating the parsimony of ALV).

```
Inputs: home_image = Image at home position
        current_image = Image at current position

home_landmarks = feature_detector( home_image )
home_ALV = mean(home_landmarks)

current_landmarks = feature_detector( current_image )
current_ALV = mean(current_landmarks)

homing_vector = home_ALV − current_ALV
```

ALV obviously has a heavy reliance upon the landmark detection. In Lambrinos et al. [2000], it was tested successfully - however, this was in a virtual arena where the landmarks were precomputed. In this arena it worked well, and was robust to the random insertion and deletion of landmarks. However in real life systems it is dependent upon the feature detection step performed on the input images. This must be necessarily be robust to changes of scale - landmarks will be different distances from the robot when it is at different points in the arena, but must be detected equally. This was not modelled in Lambrinos et al. [2000] - the disruption in landmarks was even, and not affected by apparent size.

Biologically, however, feature (and hence landmark) detection is one thing that eyes are optimised to do. Thus the requirement for a feature detector is, like a visual compass, not necessarily a barrier to it being an accurate model of cricket homing. However, since the algorithms we have for detecting features and landmarks are not as good as those found naturally, it is still a problem in a robotic model.

2.1.1.1 Implementation

To calculate the ALV, `alv.m` calculates a landmark vector from both the home and the current image, and then subtracts the home vector from the current vector. The landmark vector is calculated by `im2vec.m`, which passes the input image to a feature detector (resized appropriately), and then uses the column that each detected feature is detected in to calculate a vector direction.

Corner and feature detectors were used rather than pure landmark detectors as they perform very similar functions. Instead of writing a feature detector of my own, previously published ones were used. These were downloaded as MATLAB functions.

9 point FAST corner detection [Rosten and Drummond, 2005, 2006] To speed up the processing time, images were resized to be only 100 pixels wide before the detector runs.

Harris edge detector [Harris and Stephens, 1988] This is a well establish standard edge detector.

Loy and Zelinsky's Fast Radial Feature Detector [Loy and Zelinsky, 2003]

Scale-Invariant Feature Transform (SIFT) [Lowe, 1999] This, as the name suggests aims to identify features despite the scale they appear at. This would appear to be ideal for a function that hopes to identify the same landmarks at varying distances from them. Unfortunately, this detector often detects more landmarks at a particular location if that location is closer. This is a natural consequence of the resolution of the camera limiting identification of landmarks, but causes this detector to perform worse than expected.

A biologically accurate model of a crickets eye This was implemented by Michael Mangan. This reduces the input image to something resembling the vision of a cricket, then further reduces it to a one dimensional strip. Boundary lines on this one dimensional strip were found and used as landmarks. This was the method used in Lambrinos et al. [2001]. This is optimised to work where highly visible landmarks crossing the horizon line have high contrast against a relatively featureless background.

Where these had variable options, they were used with the default options. This was done, because with such a range of feature detectors, and such a range of options, it was infeasible to vary them manually. Applying a parameter optimisation method was deemed outside the scope of the project.

None of these detectors perform particularly well on a natural scene. (An example of this failure can be seen in Figure 2.1.1.1) For any of these to perform each point must estimate a landmark to be at a relatively consistent place. A level of inaccuracy is acceptable, and even occasionally useful (It can help perform obstacle avoidance). However, it should be noted that this environment has a relative paucity of prominent landmarks, which would lead to these feature detectors performing badly.

In the final trials, the Harris detector was used, as it was relatively lightweight yet performed as well as all the others.

Lambrinos et al. [2001]'s detector was also used. This seems optimised for highly visible landmarks, possibly moved between. This matches up to only one of the conditions. It is therefore expected that it would do best on this.

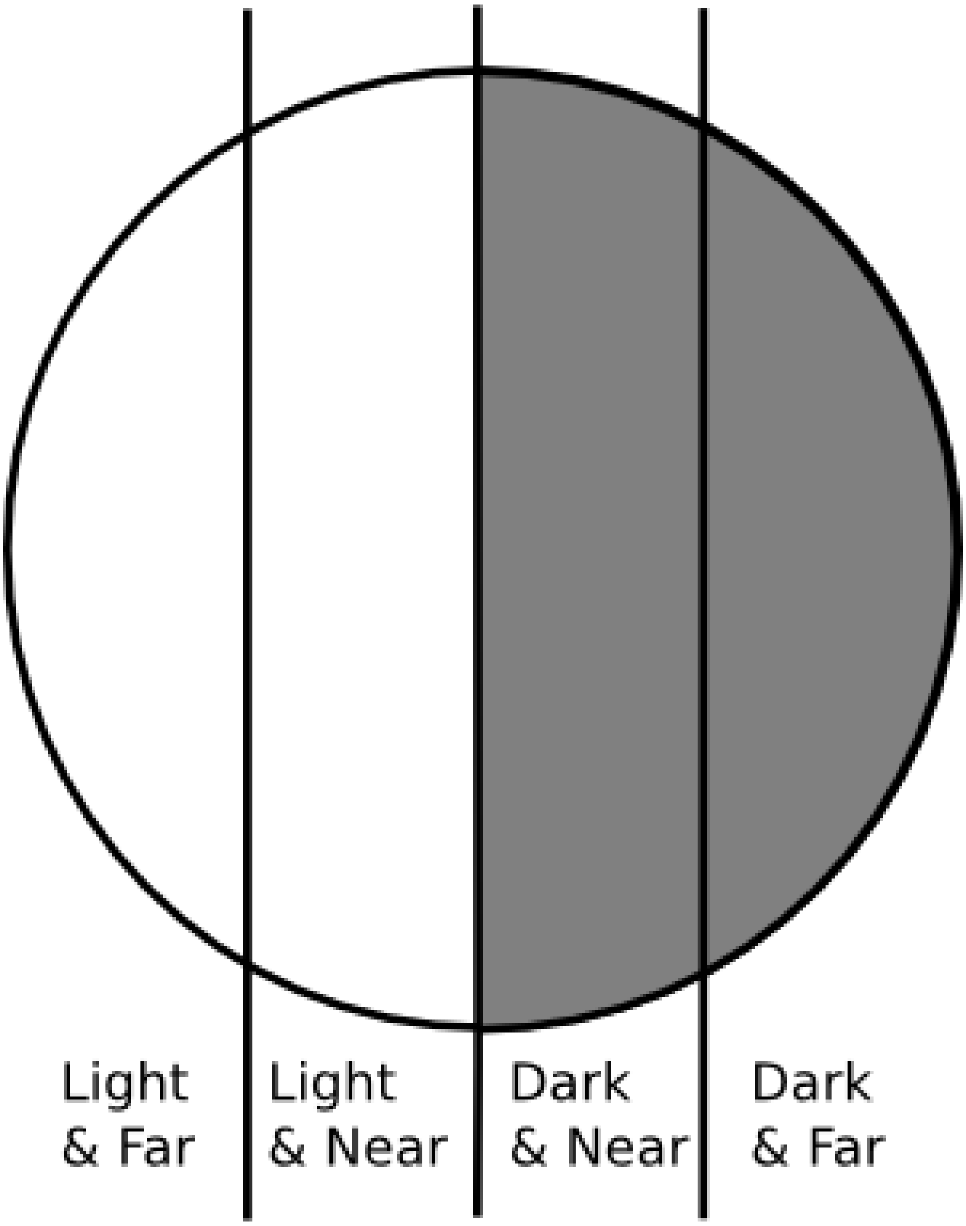

Figure 2.1: Performance of ALV on Natural Scene. The black arrows represent the homing vector. The background colour represents the error in the end vector (red is worse, blue is better). The red arrows represent the Landmark Vector.

2.1.2 Warping

Warping [Franz et al., 1998] is a more computationally expensive method. This consists of deforming the current image in such a way as to resemble movement with respect to the current position. Whichever of these most closely resembles the image at the home location is chosen. This process is then iterated until the robot has navigated to the home location.

```
Inputs: home_image = Image at home position
        current_image = Image at current position

reduced_home_image = reduce_image(home_image)

for all values of rotation in rotations, direction in
                        directions and distance in distances

        reduced_image = reduce_image(current_image)
        warped_image = warp_image(reduced_image, rotation,
                         direction, distance)

        image_distance = difference(warped_image,
                        reduced_home_image)

        if image_distance is smallest yet seen
                best_movement = [rotation, direction, distance]
        end

end
```

The process to warp the images is performed by `warp_image`. This performs the following algorithm

$$tan(\psi + \delta) = \frac{\rho sin(\theta - \alpha)}{1 - \rho cos(\theta - \alpha)}$$

where α is the direction of the homing location, ψ is the rotation of the image, θ is the original location of the pixel, ρ is the distance, as described above, and δ is the final location of the pixel.

The reduced image that warping works on is a 1 dimensional strip taken from around the horizon. In this it is similar to the some implementations of the Average Landmark Vector, as described above. It therefore has the same limitations in terms of using only landmarks that are present on the horizon, and working best on readily distinguishable landmarks. It, however, is not exactly the same because it can work on more than binary data - it can use the grayscale produced

Figure 2.2: A sample 1-dimensional image used by warping in the `3 Landmarks` dataset

by a gradient change, rather than a step change as a landmark. In this respect it is more tolerant of naturalistic scenes than ALV. It can use pixel as a landmark, and not just those which exist upon explicit discontinuities or features.

In this it shows its difference from those algorithms which explicitly match landmarks - warping can match areas of varying colour which do not have the contrast to be detected as distinct landmarks. Nevertheless, warping does attempt to match the position of landmarks in the current image with landmarks in the home image.

Some more advanced implementations of Warping operate upon a 2-dimensional image [Möller, 2009]. These show gains in performance upon the 1 dimensional algorithm. Due to the increased complexity and lack of substantive qualitative difference in performance from the 1-dimensional algorithm, these were not implemented.

It is not necessary to perform computationally expensive exhaustive global search to perform warping. It is possible to use more parsimonious search techniques(It is worth nothing that the search space is not without local minima). For simplicity, however, I have used global search.

Warping makes an assumption that all landmarks are an equal distance from the current position of the robot. While this will obviously not remain true, it is robust to violations of this assumption. This contrasts however, with the assumption made by ALV and Gradient Descent that there is an even distribution of landmarks. In both captured sets of images, but especially those captured in the arena, all the landmarks are located in a ring around the outside of the arena. This more closely conforms with Warping's expectations of landmark distribution than ALV's or Gradient Descent's. Since all three are robust to this assumption being violated, this may not matter significantly to the relative performance.

It is commonly thought that Warping is less biologically plausible than other, more parsimonious algorithms. This is due to the necessity of storing a representation of the entirety of the scene at the home location, and the high computational complexity of calculating all the combinations of rotations, distances and directions.

2.1.2.1 Implementation

`warping.m` begins by reducing both the current and home images to a small strip using `reduce_for_warping.m`, described below. It then iterates over 32 possible directions and 10 different distances (evenly spaced), and for each combination calculates the warped image. These are then compared against the input image by using the sum of squared errors. Finally, the values which produce the smallest distance are read off, and returned. This continues until the robot has reached the home location (a stopping threshold was not defined, as the position is known externally when using the gathered datasets)

The other algorithms I have been comparing Warping against assume that rotation is constant - and the images I provide have a rotation of zero. To provide a level playing field and to make my implementation more efficient, I have not iterated over rotation of the images themselves (though this was initially implemented, it is now commented out). Before this was done, performance was only slightly worse than it is now -only in a few positions did the algorithm think that the correct orientation was anything other than rotated.

Images are compressed into a single strip by `reduce_for_warping.m`. This performs Gaussian convolution upon the images, then selects the strip of pixels along the midway point of the images and uses only them. This line of pixels is then reduced to a fifth of their size, in order to speed up the process of warping them. Ironically, it is this preparatory process that takes up the most time. The Gaussian convolution is performed in order to remove any transient landmarks from the scene - such as noise and local detail.

`warp_image.m` uses the warping equation described in §2.1.2. It uses this to calculate the displacement for every pixel position in the original. This shift, together with the pixel values are then passed to `interp1`, an inbuilt MATLAB function that interpolates the pixel values. This produces a output of the same length as the original input image, with an average taken of points where the image has been compressed, and interpolation performed on points where the sampling frequency of the warped pixels is less than the sampling frequency of the output pixels. This was done in order for the image at the home location to be identical in format to the warped image. This allows easy computation of the difference function

Since the values used in `warping.m` are fixed, it would be possible to use a lookup table to speed this section of the code. This optimisation was not done because the code presented here is not particularly slow running in MATLAB.

2.1.3 Gradient Descent

Gradient Descent, or Descent in Image Distances, first developed by Zeil et al. [2003] is based upon that observation that the view from near to the home location is more similar to it than the view from further away. This similarity measure is measured in the Root Mean Square (RMS) of the Sum of Squared Errors (SSE):

$$\sqrt{\sum_{p=1}^{n}(i_p - j_p)^2} \tag{2.1}$$

where n is the number of pixels in the image, i is the home image, and j is the current image.

To use this metric of similarity a gradient descent method must be used. There are many of these: One of the simplest is "RunDown" [Zeil et al., 2003]. This moves in a random direction until the gradient starts decreasing, and then changes direction. (This is similar to the Conjugate Gradient Method of gradient descent, where the difference surface is a mathematical function of the position.) A more complex model of this has been performed by simulating the chemotaxis performed by C. elegans. [Zampoglou et al., 2006].

The "Triangular" gradient descent method [Zeil et al., 2003] takes samples from three separate positions, and from this triangulates the local gradient. By travelling in the steepest direction down through this gradient, an accurate vector towards the home location can be obtained. This is an approximation of the Steepest Descent Method [Mitchell, 1997]: instead of calculating the gradient for the current position, an estimation is obtained by triangulating from three local points.

A further development on this technique is the Matched Filter Descent In Image Distances(MFDID) of Moller et al. [2006]. This approximates sampling several sample points infinitesimally close to the current image using similar techniques to the Warping method of Franz et al. [1998]. This has advantages in not requiring movement in order to obtain a non-random direction of travel. This has more ecological validity, because small exploratory movements have not been observed in crickets (or other insects) when they are attempting to home.

2.1.3.1 Implementation

Due to its superior performance and conformity with the inputs and outputs passed to the other homing methods, the "Triangulated Descent" method of Zeil et al. [2003] was implemented. The main function `gradient.m` takes four images: a home image, an image from the current location, and two image from specified offsets from the current location. It finds the RMS error of each of the the local images compared against the home image by using the square root of `sse.m` (This calculates the Sum of Squared Errors for two arrays of equal size).

`triangulated_descent.m` then calculates a bearing by taking the three RMS values and associated offsets as points in 3-dimensional space defining a plane (using `cross(rms_vec1-rms_vec2, rms_vec1-rms_vec3)`). This defines the plane in terms of the normal, which, when reduced to its x and y components becomes a vector towards the direction of the steepest descent of the gradient surface [1].

The intensity of the gradient of the slope (as measured by the z co-ordinate of the normal) can be used as a crude estimate of the distance to the home location.

The two points other than the current location used for triangulation have been selected using two schemes. One, simpler, scheme merely samples two adjacent points in orthogonal directions. While this functions well in the image datasets, it does not function well as a model of real performance in the robot or the cricket. Each time a cricket decides what course to take it does not shift in order to obtain two other viewpoints.

I have therefore also implemented another scheme (`gradient_path` whereby the last two points that extend in two orthogonal directions are used as directional cues. When the run is started, the agent moves in a small right angle, and henceforth uses previous image locations to triangulate and estimate the difference gradient at their current location. [2]

2.2 Datasets

The easiest way to test the algorithms is to model their behaviour against datasets of precollected images. This allows testing of the algorithms against fairly naturalistic image data, while still retaining the ease of scripting repeated trials that computers allow.All four of these sets of data were captured in a panoramic view: a camera faced upwards into a hyperbolic mirror (or ball bearing), and hence the entire scene was captured.

In development, they were tested against a freely available set from the University of Bielefeld[Vardy and Moller, 2005]. This was captured across the floor inside a compting lab, and so is relatively naturalistic. There are 10 by 17 images offset by 30cm each (leading to an area of 2.7m by 4.8m being covered). This will be referred to as the `bielefeld original` dataset.

---

[1] There are, of course, two normals to the plane. The correct one was was returned by inverting the normal if the normal travelled from above the plane to below it.

[2] This caused a large number of complex conditions to be added to catch repeating loops and store previous image data, which is done in `walk_path.m` (§2.4.1)

Also used were captured images from the cricket arena. These were captured by Michael Mangan. These were captured at distances of . These were taken in four conditions:

- The natural background used for the cricket trials. (`Natural Scene`)
- A scene with no visible markings on the walls, just smooth white walls. Some slight light gradients are visible.(`Blank Walls`)
- A scene with 3 large black landmarks visible on the walls, which are otherwise blank (`3 Landmarks`)
- A scene with 2 large black landmarks visible on the walls, which are otherwise blank (this was not used)

To unwrap the Bielefeld data, a modification of a GUI tool (`omni_unwrap.m`) for unwrapping panoramic images was used [Mabius and Tang]. The parameters we obtained by specifying the center of the images, and the radii forming the vertical upper and lower bounds of the image. This was done for every image in the dataset as a preprocessing step, to speed the performance of the algorithms. To unwrap the data obtained from Michael Mangan, his code was used: this had used external reference to calculate the height of every distance from the centre, and then iterated through these, reshaping a circular ring of pixels as a row of pixels. This, not guessing at the distortions of the image, was more accurate. This was not done for the Bielefeld data, because there was no access to the camera used to capture the data, and because the output of `omni_unwrap2.m` appears sufficiently undistorted. All three algorithms used are robust to small distortions from unwrapping: warping uses only a small slice of the image, ALV operates either only on a small slice, or on found features (which are themselves robust to small distortions) and image gradient merely requires that the distortions are consistent (which they would be).

This panoramic view closely approximates a cricket's field of view.

2.3 Robot

While the image datasets allow more experiments to be carried out in a controlled fashion, they lack certain features of the real world. Most notably, the sampling of positions within the arena is limited to a grid. As a result, it is only possible to move in one of four directions. This is obviously not ecologically valid, and no doubt has a detrimental effect on homing performance. In addition, the datasets do not suffer from transient noise introduced by the camera, distortions in the light, have no problems with the misalignment of images. All of these things can be modelled, but it is easier to test the code upon an actual robotic platform.

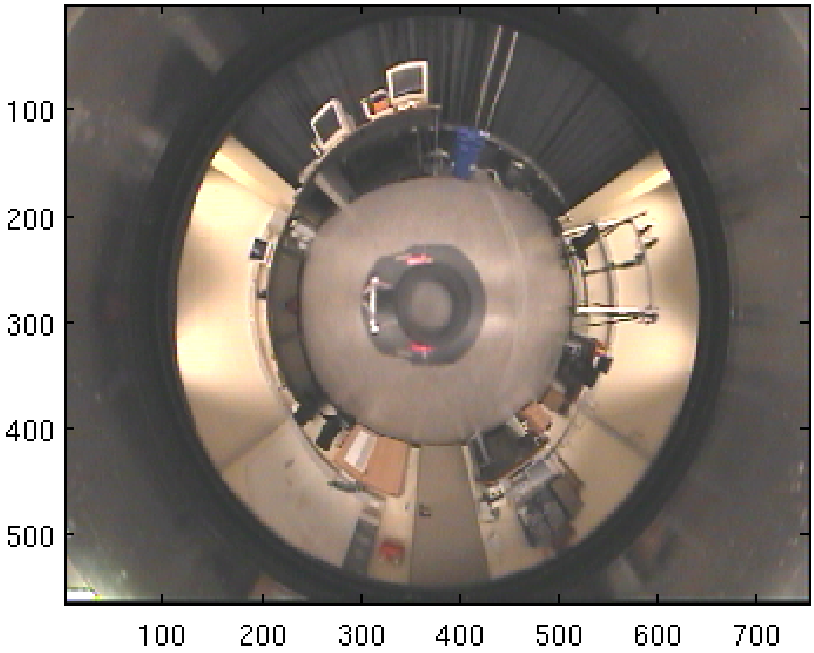

Figure 2.3: The captured panoramic image (from `bielefeld original`)

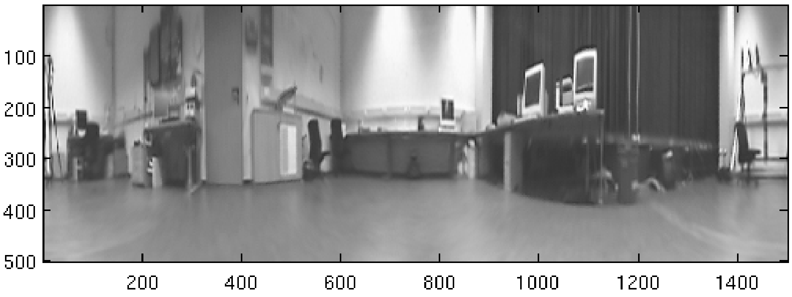

Figure 2.4: The image after unwrapping

This is useful if for no other reason than it exposes sources of error that one would not necessarily have thought to model. Since my implementation of the homing algorithms takes a sufficiently modular input and returns a sufficiently modular output, I have decide to implement adapt the algorithms to work upon a robotic platform.

The robot used was a Khephera II. This used the wireless camera system that was used to capture data by Michael Mangan. The image setup is therefore very comparable with the `Natural Scene` dataset.

This robot was run inside the same arena that the crickets were tested in. The overhead webcam could then be used to track its performance at the homing task.

2.3.1 Implementation

The implementation was fairly simple. An unwrapped image taken at the target location(the image has to be taken at the same orientation as the start position due to the lack of visual or other compasses) is passed to `run_homing.m` The following loop is then entered:

1. Get current image
2. Unwrap current image
3. Perform homing using the currently selected algorithm
4. Turn the the amount necessary to orient the robot towards the home location
5. Rotate the home image by the same amount (to keep both current and home images oriented)
6. Move forwards a set distance
7. If the proximity sensors are below a certain threshold, move away from the obstacle (`detect_walls.m`
8. Pause - this gives the camera time to recover from the noise caused by the motors.
9. (Save the quantity of rotation)

While the basic “get image / find direction / move” cycle is obvious, there are several additional features that were added due to iteration and testing. The rotation of the home location to match the current image was done to avoid the necessity for a complex visual compass system. This system relies upon dead reckoning. Due to the small size of the arena, this appears sufficient. Rather that rotating back to the original orientation, it was decided to avoid errors caused by dead reckoning drifting out of alignment by excessive movement[3] by rotating the home image, This has the advantage that it is non-destructive, meaning it can be done upon every iteration of the loop to keep track of orientation implicitly.

`detect_walls.m` was added after observing that the homing algorithms, when not performing at their best, could navigate the robot towards the walls of the arena, whereupon the robot would collide. This step merely moves the robot forwards or backwards to allow it to turn freely upon the next iteration through the loop.

One issue that seemed to be persistent was a failure of the signal from the video feed. This was eventually traced to the Khepera still attempting to move in order to increment its steps by the correct amount. This continual straining produced a voltage drain enough to completely ruin the quality of the video feed.

2.4 Performance

Performance was evaluated using the metrics Rate of Return (RR) and Average Angular Error (AAE), which were calculated from the traces generated in `metrics.m`. These are the same metrics that were used in Pons et al. [2007]. Rate of Return is the proportion of homing attempts that are successful, and Average Angular Error is the difference between the correct bearing and the bearing returned.

Due to lack of time, there was no systematic evaluation of the performance of the robotic implementation, although it did appear to have some success in homing using Warping with the Natural Scene background. The dead reckoning was tested and appeared to be constant for at least 10 iterations of movement.

2.4.1 Methods

`metrics.m` iterates over the entire given dataset, attempting to home from every point to every other point using a given algorithm. To do this, it calls `walk_path.m`. The function `walk_path.m` iterates from a given start location, attempting to reach the home location using a given homing algorithm. The bearings taken at each step, and whether the run was successful (or how it failed: did it become circular, come off the edge of the arena, or attempt to reference a

---

[3] This is also the reason that `kturn.m` turns clockwise or anticlockwise, in order to minimise rotation.

	Warping	ALV	Gradient Descent	Gradient Descent (prev. locs.)
`Natural scene`	1.35	1.51	1.0253	0.90
`3 Landmarks`	1.54	0.89	0.94	0.95
`Blank walls`	1.45	1.45	1.24	1.22

Table 2.1: Mean Average Angular Error (in radians) for combinations of datasets and algorithms

	Warping	ALV	Gradient Descent	Gradient Descent (prev. locs.)
`Natural scene`	0.0357	0.0170	0.1613	0.3248
`3 Landmarks`	0.0147	0.0872	0.1753	0.2789
`Blank walls`	0.0213	0.0248	0.0438	0.0742

Table 2.2: Mean Ratio of Return for combinations of datasets and algorithms

point which was off the edge of the arena) are returned. It is in `walk_path.m` that the `gradient_path` variant of the Gradient Descent algorithm is implemented.

In order to visualise the performance of the algorithms, `uigraph.m` was used. This shows, for any point, the bearings to home and the degree to which these vary from the correct bearing. It also shows an estimate of the distance to the home location, as well as the Landmark Vector for ALV, and the RMS difference surface for Gradient Descent.

2.4.2 ALV

It was found that the performance of ALV was dependent upon the landmark detector. This had to return a consistent vector to a single "Virtual Landmark" (VL) from all points within the arena for ALV to perform well, which neither "harris" nor the Lambrinos et al. [2001] detector did on most datasets. (Exploratory testing suggests that none of the other detectors performed better).

Often there was a distinctive pattern of homing, where correct vectors would be generated, but only from two opposing directions(Figure 2.4.2). This resulted from the landmark vectors being consistent, but failing to change in intensity with the distance to the virtual landmark. With only the direction and not the distance to the landmark known, it is impossible to locate the home target more accurately than upon a single line.

The dataset where ALV functioned best was on `3 Landmarks`. While the vectors to the Virtual Landmark were inconsistent, they were sufficient to allow the algorithm to home. This homing was sometimes unreliable when approaching from certain directions and nearby to the homing location. It could be expected that the best functioning dataset would be `3 Landmarks` due to its clear and

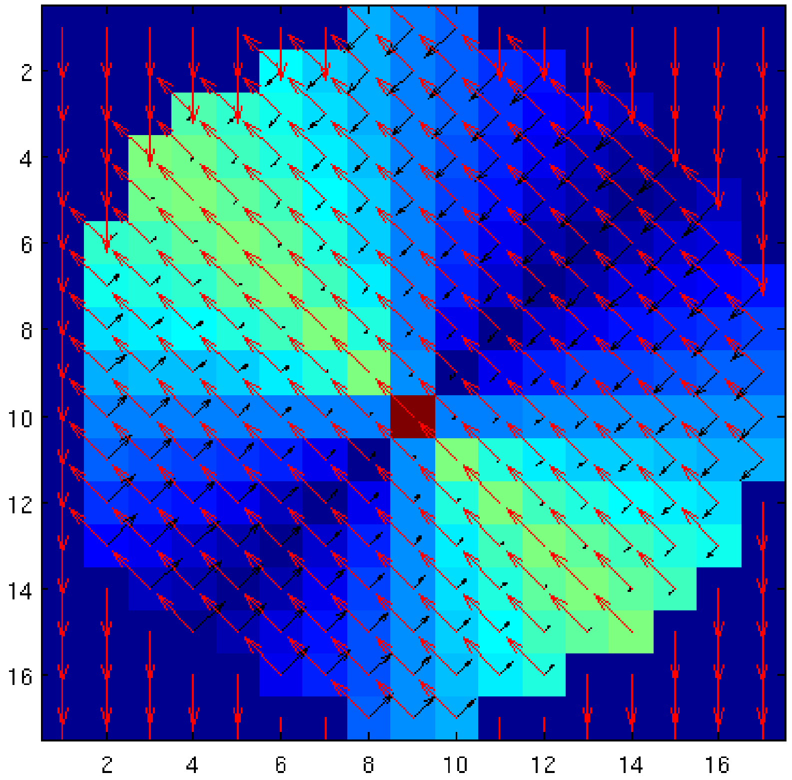

Figure 2.5: ALV attempting to home to (10,9) when used with an artificial set of unit vectors towards a Virtual Landmark at (-1000, - 1000). The red arrows indicate the vectors towards VL, the black arrows indicate the estimated route to home. The background colour indicates the error in estimated bearing.

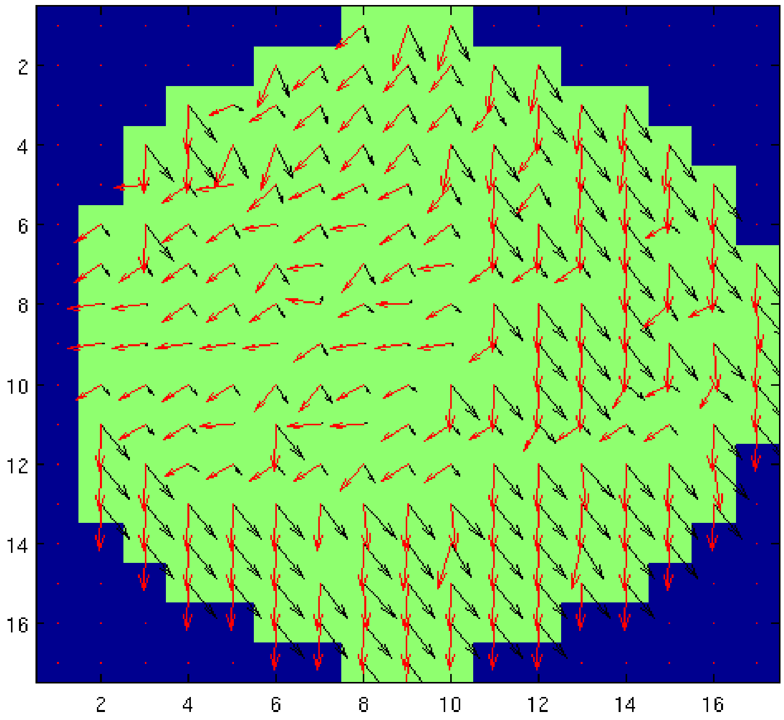

Figure 2.6: ALV when used with `Blank walls`. The red arrows indicate the vectors towards VL, the black arrows indicate the estimated route to home.

consistent landmarks. Interestingly, however, the Lambrinos et al. [2001] detector still did not perform well within the arena, despite being optimised for such environments. It finds a uniform non-converging field in a single direction, which may result from one of the landmarks being more attractive than the others. It was the case in all datasets that the Lambrinos et al. [2001] detector performed worse than "harris", and therefore only "harris" is shown. In `Blank walls` and `Natural Scene`, several patches can be seen. (Figure 2.4.2)These patches have very distinct VL, and hence homing doe not work between them. This will naturally be a failing of ALV in larger scenes - after a certain distance, a new set of landmarks will be seen which will sum to a new Virtual Landmark. Whether this results in a discontinuity or a more subtle change, ALV will be unable to home.

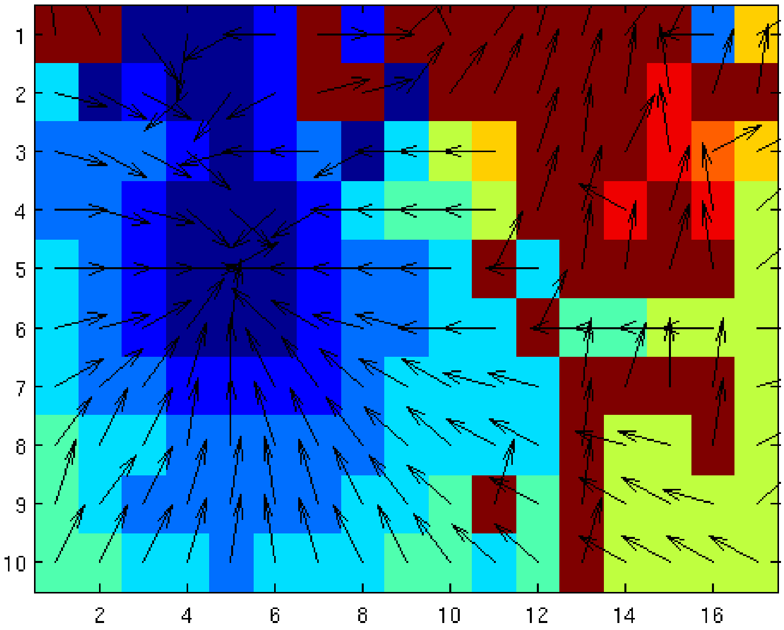

Figure 2.7: Warping when used with `bielefeld`. The black arrows indicate the estimated route to home. The background colour represents the error in the end vector (red is worse, blue is better).

2.4.3 Warping

On the Bielefeld dataset, the Warping algorithm performs well for a proportion of the arena, beyond which it breaks down. The algorithm indicates a smoothly increasing distance metric (ratio of distance to landmarks to distance to home location) for the areas where the algorithm is able to home, and a consistent "1" in areas where it is not. I would assume that the inability to home over larger distances would be a lack of sufficient resolution in the reduced images that warping operates on. This is combined with the limited set of bearings warping could use (16 in total), which become increasingly coarse-grained as the ratio increases.

Within the arena, Warping does not fail as often due to the distance being too large, as it does on the Bielefeld dataset. It does however, continue to show errors near the edge of the arena. This is more acute than with the Bielefeld dataset, which can presumably be attributed to the arena wall taking up such a large proportion of its field of view. In the Bielefeld dataset, the landmarks surrounding the field of capture are usually further away (and where they are not, the algorithm fails in a similar fashion).

Warping is very successful in the `Natural Scene` dataset, failing only occasionally, even when homing to a location that is right next to the arena walls. It also correctly homes even when the ratio assigned is "1".

Performance is degraded in the `Blank walls` dataset. While there are few cues, warping is able to use subtle light cues to it's advantage.

Performance is further degraded with the `3 Landmarks` dataset, with more false matches being made. This can be presumed to be due to there being fewer details in order to make matches: and what details there are are ambiguous, being perceived as a series of stark black marks for all landmarks. Landmarks are matched by their grayscale luminosity, relative width and relation to other landmarks, and in this condition, one of those cues is absent and another (relation to other landmarks) is weakened due to the sparseness of landmarks.

2.4.4 Gradient Descent

In the Bielefeld dataset, Gradient Descent works well across most of the arena. Within a very close range, it is extremely accurate, as the RMS difference surface increases sharply. Further away, it begins to plateau, with local minima out towards the edges. These, naturally, produce false bearings. (Figure 2.4.4)

Gradient Descent performs less well on the `Natural Scene` dataset, because of the increased level of noise. This noise is likely to be part of the dataset (changing light levels, etc), and not embedded in the scene in reality (of course, sampling live data has more noise). There is however, a gradient, if rough, which decreases towards the home location. A function that samples multiple local points and uses an average to produce a vector heading would likely be more accurate.

Performance is degraded in `3 Landmarks`, a product of the less well defined gradient, both in terms of intensity and noise. The relative paucity of cues resulting from the walls being blank apart from the landmarks causes this. Similarly, performance is even worse in the `Blank walls` condition, where the noise completely overwhelms any consistent gradient in the difference surface.

When traversing the arena, two different algorithms are used. One takes two immediate offsets and calculates the gradient based upon these. The other is more plausible in that it uses images from previous locations to calculate the current gradient. The second consistently outperformed the first. This may not be due to an innate superiority: while paths were eliminated if they formed loops, this algorithm was was only eliminated if it started repeating itself (which took a longer time to achieve). However, its increased performance may also have been attributed to its ability to avoid getting caught in local minima: its reference points could not be closer than the naive implementation.

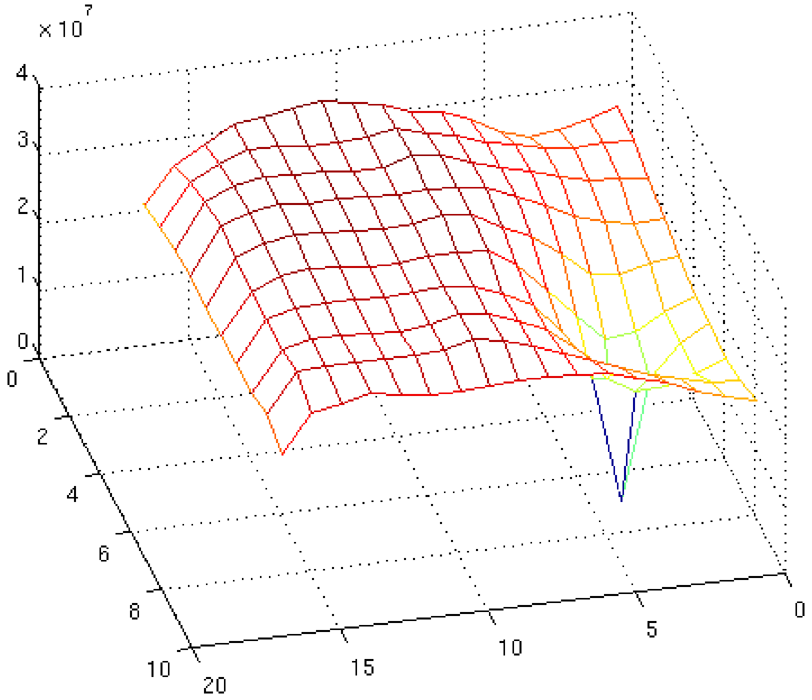

Figure 2.8: The RMS difference surface for `bielefeld original`. The home location is (4,7)

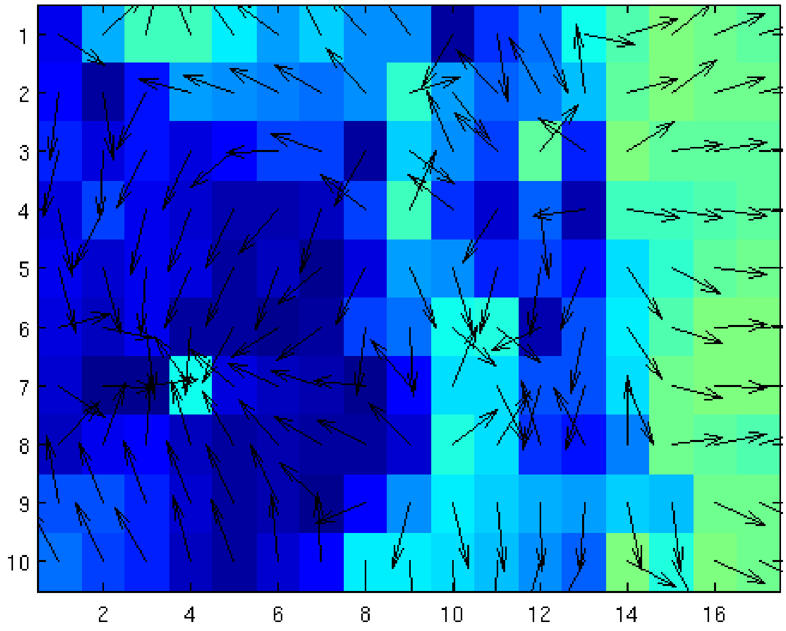

Figure 2.9: The homing vectors for Gradient Descent on `bielefeld original`. The home location is (4,7). The background colour represents the error in the end vector (red is worse, blue is better).

2.5 Discussion

2.5.1 Algorithms for use in robots

None of the algorithms tested could reliably return to the home location. While performance could be improved with optimisation, this suggests that non of these algorithms could be reliably deployed in a “real-world” situation.

Gradient Descent performed best in these trials. The main weakness of gradient descent was it’s susceptibility to noise. It depends upon the RMS difference surface of captured images being smooth, and noise can violate that. By taking several samples at each point, this could be mitigated. Additionally, a more intelligent descent strategy can be used in order to minimise the effects of local minima.

Gradient Descent and ALV are both relatively lightweight algorithms. ALV, for instance, has been implemented entirely in analog circuitry upon a robot [Möller]. ALV, however, only performs well when landmarks are easily identifiable (and depending on the identifier, cross the horizon). For more generalised natural scenes, gradient descent would be a better choice.

Warping has scope for optimisation, however: most time is spent exploring the search space of rotations, distances and bearings, which could well be optimised. One possible strategy would be to explore the search space very coarsely, and then do a more fine-grained search in more promising areas.

One weakness of both Gradient Descent and ALV is their need for an external cue to inform them of rotation. This could be mitigated using a compass system derived from the visual data §2.5.2.

Performance of a robot will also depend upon the distribution of landmarks within a scene. In all the datasets used, the landmarks surrounded the space through which the robot could move. Performance will change if the robot is moving through a field of obstacles/landmarks.

Similarly, moving through such a field, a robot would most likely have an obstacle avoidance system set up. Here ALV has an advantage in that it naturally performs obstacle avoidance when moving through fields of landmarks [Hafner, 2004]. On the contrary, warping has a tendency to move so that landmarks occlude the home location [Franz et al., 1998].

2.5.2 Algorithms as models of crickets

It can be seen that crickets outperform algorithms considerably. While crickets do show variance between light conditions, in all three conditions their performance is similar and robust. In keeping with the models, a brief exposure (a single trial) is enough for the crickets to gain knowledge of its location.

When comparing algorithms, a clear distinction can be made between both Warping and Gradient Descent, which has a preference for the `NaturalScene`, degraded performance with `3 Landmarks` and poor performance with `Blank Walls` and between ALV, which performs best on `3 Landmarks` and worse on the others. Since crickets showed best performance with `NaturalScene`, Warping and Gradient Descent appear to match better.

There is a discrepancy, however, between the performance of crickets and algorithms on `3 Landmarks`. The decreased performance of crickets may be explained by the crickets attraction for the landmarks due to the scototaxic instinct mentioned in §1.4.3. The algorithm, being focused only upon homing, does not have this distraction and can perform better. The crickets performance in the `Blank walls` condition was attributed to a subtle light gradient in the arena [Wessnitzer et al., 2008], as well as perceiving the borderline between the wall and the floor. It may be that the sensing apparatus is not as sensitive as a cricket to such subtle differences. In addition, the cricket is not restricted from using its view overhead and the height of the floor. Warping, taking a narrow strip around the horizon, is denied access to these cues. Gradient Descent is more able to use these cues effectively. Subtle differences in lighting are overwhelmed by noise in capturing each image, and the crack connecting wall and floor is a very small proportion of the total image.

This reliance upon landmarks which cross the horizon can be mitigated in Warping by its extension to a 2-dimensional image (as done by Möller [2009]). This would make Warping more realistic.

While the datasets have their alignment guaranteed by virtue of being collected in that way, the crickets and the robots do not. While the robot is able to use dead reckoning over the short distance it has travelled, it seems unlikely that crickets would merely keep track of their orientation this way (if for no other reasons than that they are not in continuous contact with the ground, unlike a robot). Instead, some kind of tracking by optical flow seems likely . While only warping supports non-aligned images, the gradient descent technique can be adapted to align images taken at varying locations. Zeil et al. [2003] found that "Rotational difference functions change little with distance from the reference location." - hence the current image can be oriented into a consistent rotation by finding the global minima when the current image is rotated with respect to the home image.

This can be used as a preprocessing step for ALV or Gradient Descent.

2.6 Evaluation

`walk_paths.m` is more sensitive to failure conditions than crickets are. When a cricket comes against the edge of the arena, it usually follows it round. When `walk_paths.m` comes against the edge, it gives up in failure. The same happens if it starts repeating itself: a condition a cricket is unlikely to get into.

Performing `metrics.m` upon the `bielefield original` dataset would allow comparisons to be drawn against other implementations of the same algorithms, and would allow comparison of the arena dataset against that of the arena.

It would have been preferable to conduct systematic testing of the algorithms with the robot. The implementation was done, and the robot appeared to perform homing with Warping in the `Natural Scene`, which was expected to be the best performing algorithm and condition. Testing in real robot platforms is a better proof of concept than in even realistic datasets because no element can be abstracted. The problem of orientation, which was ignored in trials upon the datasets, was forced to be solved for trials with the robot.

In some respects, however, the robotic platform is easier. Rather than being forced to rotate at 90 degree increments, it can perform arbitrary turns. This allows it to travel a lot more directly towards the target. Since many of the algorithms can be inconsistent from one location to the next, this is one reason that the RRs are all so low. Again, the low tolerance for failure in `walk_paths.m` would not be present: loops and contacting the edge of the arena would both result in continued attempts.

The easiest way to improve the performance of the algorithms would be to tune the parameters they accept. This could be done using, for example, genetic algorithms to find the optimal size of step to vary bearing and distance in Warping, the size of the one dimensional images used and the amount of smoothing performed upon them. These are currently chosen due to not imposing a huge overhead when processing while still appearing to work well.

At the cost of increasing the overhead in performing Warping further, confirmation of the homing vector could be found by performing the inverse transformation upon the stored home image, and matching it against the current image. If this was done, the chance of local noise or occlusions affecting performance would be decreased, which should result in an increase in performance.

Landmark detecting functions for ALV can be tested directly: an ideal landmark detection function will have every landmark vector converge upon the same virtual landmark. This would provide a quicker way to evaluate the performance of a landmark function than running virtual trials and evaluating the actual performance.

One interesting part is that all three algorithms are capable of estimating their distance from the home location. Since their accuracy varies with distance, this information can be used to determine the likelihood that they are accurate. Since they have varying distances within which they are accurate, a Frankenstein strategy could be used, where ALV or Warping is used until the agent is close to the target location, at which point gradient descent could be used. Similarly, in robots, a less accurate, long distance modality could be used until the local visual homing system starts gaining confidence (expressed as a consistently small distance).

3. Conclusion

The performance of the algorithms was not on a par with that of the crickets. The crickets could consistently home to a set location, even with conflicting demands (such as escaping from the arena). However, given the unoptimised nature of the algorithms, and the relative crudeness of the input (snapshots every few centimetres, as opposed to a continuous stream of visual data) this should not be too surprising.

However, none of these algorithms can directly be considered to represent a cricket's process of visual homing. Warping is not considered realistic because processing all of the combinations of rotation, bearing and orientation demands a huge processing overhead, even with a more efficient search than full global search. While it was suggested this could be parallelised with matched filters, this is argued to still involve an unrealistic quantity of neurons [Vardy and Mller, 2005]. Gradient descent, while requiring less processing, requires movement before a homing vector can be calculated. ALV, while more cognitively plausible, fails to use cues such as subtle light gradients, which crickets have been shown to use for homing. Instead, it requires that landmarks be distinctively segmented from the image.

Of course, the performance of the cricket in the arena does not directly represent it's process of visual homing. Crickets are, unlike robots, not singleminded in pursuit of set aims. The crickets notably spent more time performing wall following behaviours and exploring the arena than they did attempting to find the cool spot.

Indeed, it seems possible that crickets do not have a dedicated homing algorithm at all. It is suggested by the cricket trials performed that crickets have an awareness of their location in the arena, not merely distance and bearing to a single point. Being models purely of visual homing, this is not represented well by any of these models.

However, some characteristics of a successful algorithm may be found. A successful algorithm should be robust against a range of inputs and should be miserly with them (it should not discard information unless there is a specific reason for it). It would be surprising if optical flow was not involved in this optimal algorithm - Warping is a crude exploration of optical flow, and an improved version of Gradient Descent, Matched Filter Descent in Image Distances [Moller et al., 2006] incorporates optical flow information to estimate the gradient of the RMS difference surface, and is reported to be even more successful.

Bibliography

Guy Beugnon and Raymond Campan. Homing in the field cricket,gryllus campestris. *Journal of Insect Behavior*, 2(2):187–198, March 1989. doi: http://dx.doi.org/10.1007/BF01053291. URL `http://dx.doi.org/10.1007/BF01053291`.

B. A. Cartwright and T. S. Collett. Landmark learning in bees. *Journal of Comparative Physiology A: Neuroethology, Sensory, Neural, and Behavioral Physiology*, 151(4):521–543, December 1983. doi: http://dx.doi.org/10.1007/BF00605469. URL `http://dx.doi.org/10.1007/BF00605469`.

O. Dangles, D. Pierre, J. P. Christides, and J. Casas. Escape performance decreases during ontogeny in wild crickets. *J Exp Biol*, 210(18):3165–3170, September 2007. doi: http://dx.doi.org/10.1242/jeb.004648. URL `http://dx.doi.org/10.1242/jeb.004648`.

Matthias O. Franz, Bernhard Schölkopf, Hanspeter A. Mallot, and Heinrich H. Bülthoff. Where did i take that snapshot? scene-based homing by image matching. *Biological Cybernetics*, 79(3):191–202, October 1998. doi: http://dx.doi.org/10.1007/s004220050470. URL `http://dx.doi.org/10.1007/s004220050470`.

Verena V. Hafner. Agent-environment interaction in visual homing. page 629. 2004. URL `http://www.springerlink.com/content/cetxe7ddvn06w2rf`.

C. Harris and M. Stephens. A combined corner and edge detector. pages 147–151, 1988.

Uwe Homberg. In search of the sky compass in the insect brain. *Naturwissenschaften*, 91(5):199–208, May 2004. doi: 10.1007/s00114-004-0525-9. URL `http://dx.doi.org/10.1007/s00114-004-0525-9`.

H.W. Honegger and R. Campan. *Vision and visually guided behavior*, pages 147–177. In , Moore et al. [1989], 1989. ISBN 0801422728.

Gerald Kastberger. Evasive behaviour in the cave-cricket, ¡i¿troglophilus cavicola¡/i¿. *Physiological Entomology*, 7(2):175–181, 1982. doi: http://dx.doi.org/10.1111/j.1365-3032.1982.tb00287.x. URL `http://dx.doi.org/10.1111/j.1365-3032.1982.tb00287.x`.

D. Lambrinos, R. Moller, T. Labhart, R. Pfeifer, and R. Wehner. A mobile robot employing insect strategies for navigation. *Robotics and Autonomous Systems*, pages 39–64, January 2000. ISSN 0921-

8890. doi: http://dx.doi.org/10.1016/S0921-8890(99)00064-0. URL `http://dx.doi.org/10.1016/S0921-8890(99)00064-0`.

Dimitrios Lambrinos, Ralf Möller, Rolf Pfeifer, and Rüdiger Wehner. Landmark navigation without snapshots: the average landmark vector model. In N. Elsner and R. Wehner, editors, *26th Goettingen Neurobiology Conference*, volume 1, 1998.

Dimitrios Lambrinos, Thorsten Roggendorf, and Rolf Pfeifer. Insect strategies of visual homing in mobile robots. In *Biorobotics - Methods and Applications. AAAI Press*, pages 37–66, 2001. URL `http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.30.5189`.

D. G. Lowe. Object recognition from local scale-invariant features. volume 2, pages 1150–1157 vol.2, 1999. doi: http://dx.doi.org/10.1109/ICCV.1999.790410. URL `http://dx.doi.org/10.1109/ICCV.1999.790410`.

G. Loy and A. Zelinsky. Fast radial symmetry for detecting points of interest. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(8): 959–973, 2003. doi: http://dx.doi.org/10.1109/TPAMI.2003.1217601. URL `http://dx.doi.org/10.1109/TPAMI.2003.1217601`.

David Mabius and Jonathan Tang. Technical report.

Tom M. Mitchell. *Machine Learning*. McGraw-Hill Science/Engineering/Math, March 1997. ISBN 0070428077.

Makoto Mizunami, Josette M. Weibrecht, and Nicholas J. Strausfeld. A new role for the insect mushroom bodies: place memory and motor control. In *Proceedings of the workshop on "Locomotion Control in Legged Invertebrates" on Biological neural networks in invertebrate neuroethology and robotics*, pages 199–225, San Diego, CA, USA, 1993. Academic Press Professional, Inc. ISBN 0-12-084728-0. URL `http://portal.acm.org/citation.cfm?id=169157.169186`.

Moller, Ralf, Vardy, and Andrew. Local visual homing by matched-filter descent in image distances. *Biological Cybernetics*, 95(5):413–430, November 2006. ISSN 0340-1200. doi: http://dx.doi.org/10.1007/s00422-006-0095-3. URL `http://dx.doi.org/10.1007/s00422-006-0095-3`.

Ralf Möller. Visual homing in analog hardware. URL `http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.43.9529`.

Ralf Möller. Local visual homing by warping of two-dimensional images. *Robot. Auton. Syst.*, 57(1):87–101, 2009. ISSN 0921-8890. doi: http://dx.doi.org/10.1016/j.robot.2008.02.001. URL `http://dx.doi.org/10.1016/j.robot.2008.02.001`.

Thomas E. Moore, Franz Huber, and Werner. Loher. *Cricket behavior and neurobiology / edited by Franz Huber, Thomas E. Moore, Werner Loher.* Comstock Pub. Associates, Ithaca :, 1989. ISBN 0801422728.

R. G. M. Morris. Spatial localization does not require the presence of local cues. *Learning and Motivation*, 12:239–260, 1981.

H. Nishino and M. Sakai. Behaviorally significant immobile state of so-called thanatosis in the cricket gryllus bimaculatus degeer: its characterization, sensory mechanism and function. *Journal of Comparative Physiology A: Neuroethology, Sensory, Neural, and Behavioral Physiology*, 179(5): 613–624, November 1996. doi: http://dx.doi.org/10.1007/BF00216126. URL `http://dx.doi.org/10.1007/BF00216126`.

Hiroshi Nishino. Motor output characterizing thanatosis in the cricket gryllus bimaculatus. *J Exp Biol*, 207(22):3899–3915, October 2004. doi: http://dx.doi.org/10.1242/jeb.01220. URL `http://dx.doi.org/10.1242/jeb.01220`.

J. Saez Pons, W. Hbner, H. Dahmen, and H. A. Mallot. Vision-based robot homing in dynamic environments. In *Proc. of the 13th IASTED Intern. Conf. Robotics and Applications*, pages 293–298, 2007.

Edward Rosten and Tom Drummond. Fusing points and lines for high performance tracking. In *IEEE International Conference on Computer Vision*, volume 2, pages 1508–1511, October 2005. URL `http://mi.eng.cam.ac.uk/ er258/work/rosten_2005_tracking.pdf`.

Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *European Conference on Computer Vision (to appear)*, May 2006. URL `http://mi.eng.cam.ac.uk/ er258/work/rosten_2006_machine.pdf`.

Sophie Scotto-Lomassese, Colette Strambi, Alain Strambi, Aicha Aouane, Roger Augier, Genevieve Rougon, and Myriam Cayre. Suppression of adult neurogenesis impairs olfactory learning and memory in an adult insect. *J. Neurosci.*, 23(28):9289–9296, October 2003. URL `http://www.jneurosci.org/cgi/content/abstract/23/28/9289?ijkey=e6519319b9bc1f572c`

Andrew Vardy and Ralf Moller. Biologically plausible visual homing methods based on optical flow techniques. *Connection Science*, 17(1-2):47–89, June 2005. ISSN 0954-0091. doi: http://dx.doi.org/10.1080/09540090500140958. URL `http://dx.doi.org/10.1080/09540090500140958`.

Andrew Vardy and Ralf Mller. Biologically plausible visual homing methods based on optical flow techniques. *Connection Science, Special Issue: Navigation*, 17:47–90, 2005.

T. J. Walker and S. Masaki. *Natural history*, pages 1–42. In , Moore et al. [1989], 1989. ISBN 0801422728.

J. Wessnitzer, M. Mangan, and B. Webb. Place memory in crickets. *Proceedings of the Royal Society B: Biological Sciences*, 2008.

Markos Zampoglou, Matthew Szenher, and Barbara Webb. Adaptation of controllers for image-based homing. *Adaptive Behavior*, 14(4):381–399, December 2006. doi: http://dx.doi.org/10.1177/1059712306072338. URL `http://dx.doi.org/10.1177/1059712306072338`.

Jochen Zeil, Martin I. Hofmann, and Javaan S. Chahl. Catchment areas of panoramic snapshots in outdoor scenes. *J. Opt. Soc. Am. A*, 20(3):450–469, March 2003. doi: http://dx.doi.org/10.1364/JOSAA.20.000450. URL `http://dx.doi.org/10.1364/JOSAA.20.000450`.

Appendix A. Pilot experiments

In order to settle on the light-switching paradigm, several other approaches were explored.

The Tennessee Williams paradigm attempts to replicate the well known Morris water maze paradigm [Morris, 1981], originally performed on rats. It was an initial hope that the Morris water maze could be performed directly on the crickets. If this were possible then direct comparisons between results obtained in the Morris water maze and results obtained on crickets could be made. This seemed plausible: crickets are capable of swimming, and, of course, homing using non-local cues [Wessnitzer et al., 2008, Beugnon and Campan, 1989]. To this end a pilot trial was carried out: a cricket was placed in lukewarm water with a small visible piece of cardboard jutting above the water. The cricket initially swam, but soon was content to float. Even when it brushed against the cardboard it would occasionally, but not always attempt to climb onto it and out of the water. When a plastic platform was added below the water, the cricket made no response at all.

Due to the crickets apparent tolerance for exposure to the water, attempts were made to make the water more aversive. Temperature was considered - after all, this was the aversive stimulus used in Wessnitzer et al. [2008]. Crickets, being cold blooded, become sluggish when exposed to cold, so this was ruled out. Heating the water was also ruled out, if only because it is difficult to maintain a constant temperature. Upon exposure to heat, crickets have a tendency to perish. This is the reason that Wessnitzer et al. [2008] restricted each run to a mere 5 minutes - any longer led to high casualty rates. Heat was therefore ruled out. When the water was agitated, the cricket would persist in attempting to swim. (I speculate that this is due to still water being a less dangerous environment for crickets in their natural state, whereas fast flowing water is: if there is any wind, the cricket will end up washing up against a shore, with their energy conserved, whereas in faster flowing water they may well collide against objects.) This gave me hope, however the crickets became if anything less attracted to the jutting cardboard. They would now not even attempt to grasp it when they were knocked against it. Since the Morris water maze paradigm depends upon the animals preferring resting upon a submerged platform to swimming, this was not hopeful. In addition, the waves in the waters surface produced by the agitation would expose the platform if it was sufficiently close to the surface to be useful as a platform by the cricket. Since the agitation was produced by impacts against the side of the tank, these waves also had a directional component, which would bias the results if agitation was used as an aversive stimulus.

Observing the crickets, it was noted that they had an affinity for dark hiding places. This is a common enough observation of crickets that breeder's guides note it as a requirement for them survive for even short term storage. With this in mind, a trial experiment was carried out in which the aversive condition was exposure, and the favourable condition was the crickets being hidden in a dark place (as in Walker and Masaki [1989]). To avoid the entrance being immediately visible, the crickets were placed upon a raised disc (again, made out of cardboard) with a lower section surrounding it (and beyond that, the walls). A hole was made connecting the lower section with the (obviously dark) underside of the disc. This is, of course, not visible from the starting vantage point of the cricket in the center of the disc. hen placed on the center of this disc, they moved to the edges and navigated around the lower groove. As in Wessnitzer et al. [2008], the crickets embarked upon wall-following behaviour until they came across the hole, whereupon they usually entered it. However, this process took a very short amount of time. This meant that it was diffcult to analyse their paths - there is only the short journey over the raised platform and the direction upon entering the lower groove. With such short paths to analyse, such factors as the way they were facing when placed in the arena become prominent. Observation suggested that the crickets did not appear to exhibit any learning behaviours - they merely headed towards the edge and then followed the wall.

While this paradigm did not appear functional, the idea of using light as an aversive stimuli was still good. This time, when crickets were placed in an (empty) arena, lighting conditions were varied using a desklamp held above. When crickets were upon a Post-it note placed near the center of the arena, the lights were switched off. When they moved off this spot, the lights were switched on.

While the crickets continued to mainly perform wallfollowing behaviours, once the "dark" center was found they would remain there for some time. This may have been due to a preference for the dark, or a reaction to the change in light conditions (freezing), or both. A cricket that had found the center would slowly explore the limits of the area - creeping forwards until it had emerged and the lights changed, then freezing, then retreating. This could be due to a preference for lit areas very close to dark areas (I rationalised this as wishing to remain near the dark area for purposes of shelter from threat while still remaining in the light to have warning of the threats). Alternatively, it could be simply due to an aversion to light and to changes of illumination. After some time in or near the dark, the cricket would resume exploration - this generally meaning wallfollowing.

In an effort to explore this seeming preference for "near the dark", I tried applying an aversive stimulus when the cricket was in the light but near the boundary. As with the water tank, this was agitation of the arena. My first attempts involved a horizontal jolt, which appeared to make the cricket leap away from the point of impact. When the bottom of the arena was struck, the response was not as strong.

No readily identifiable pattern was perceived in either response, and applying a consistent jolt was tricky, so this path of experimentation was abandoned.

The setup used in the pilot experiment was not ideal - I had problems switching the light in response to the cricket position. This would become even harder is there were no visible markings of the target location. Additionally, the light was positioned very close to the arena floor. This blocked the view of the camera, and provided a good additional directional cue for the crickets. Positioning the light beyond the camera would be preferable, as would not having my face visible. To resolve both issues, a computer was used to track the crickets and switch the lights on and off based upon their position. The lights were resited on a platform above the camera - this had an impact upon light intensity (discussed in §1.4.3).

Appendix B. Metrics

B.1 Average Angular Error

B.1.1 Warping

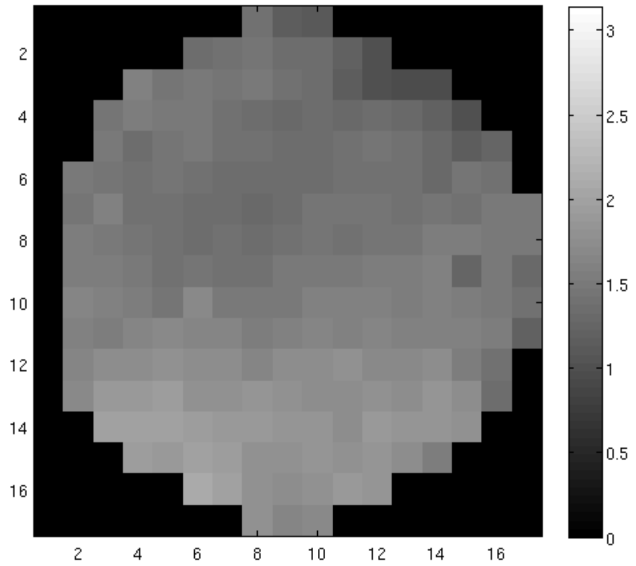

Figure B.1: 3 Landmarks - Warping

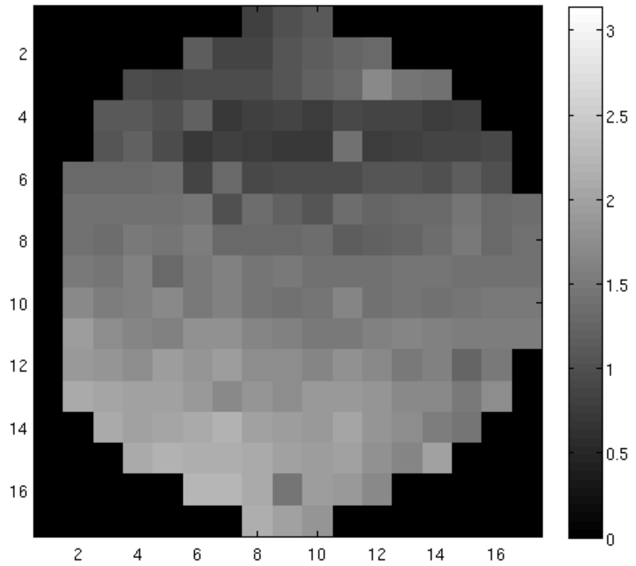

Figure B.2: Blank Walls - Warping

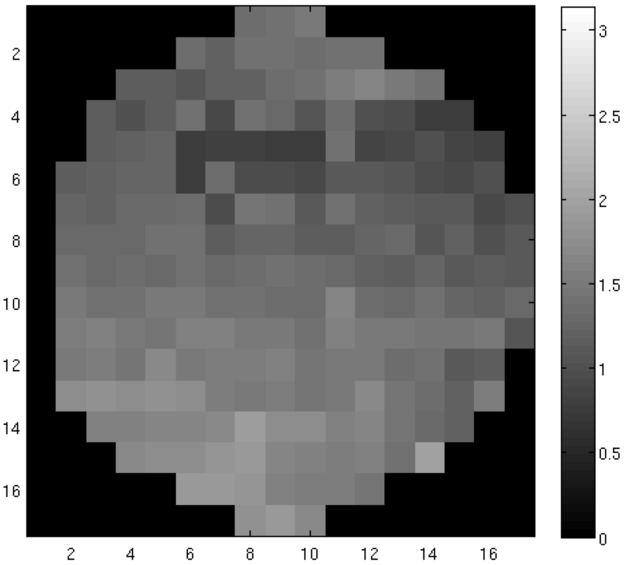

Figure B.3: Natural Scene - Warping

B.1.2 ALV

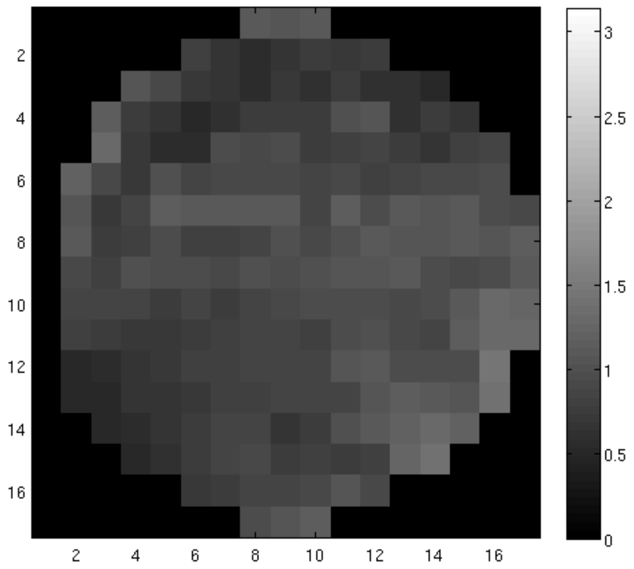

Figure B.4: 3 Landmarks - ALV

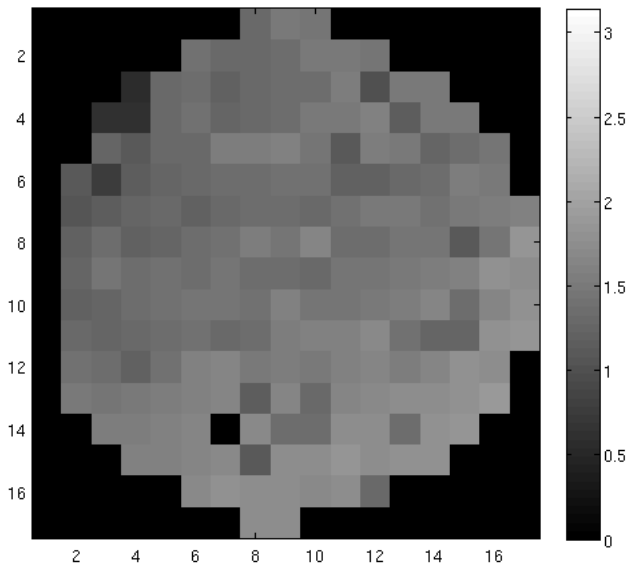

Figure B.5: Blank Walls - ALV

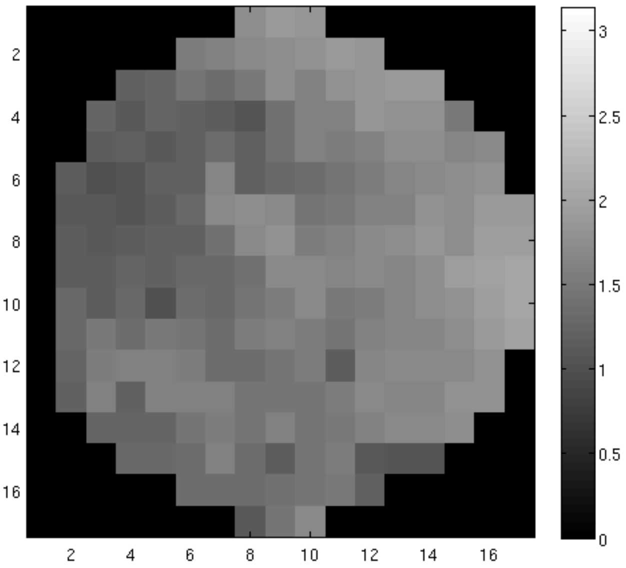

Figure B.6: Natural Scene - ALV

B.1.3 Gradient Descent

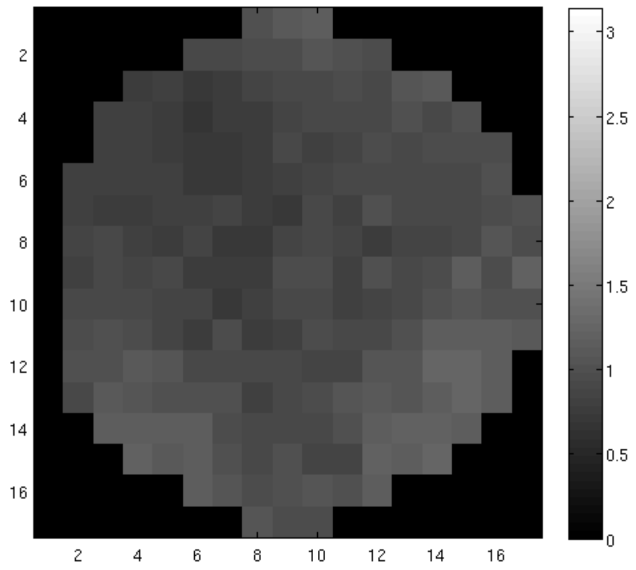

Figure B.7: 3 Landmarks - Gradient Descent

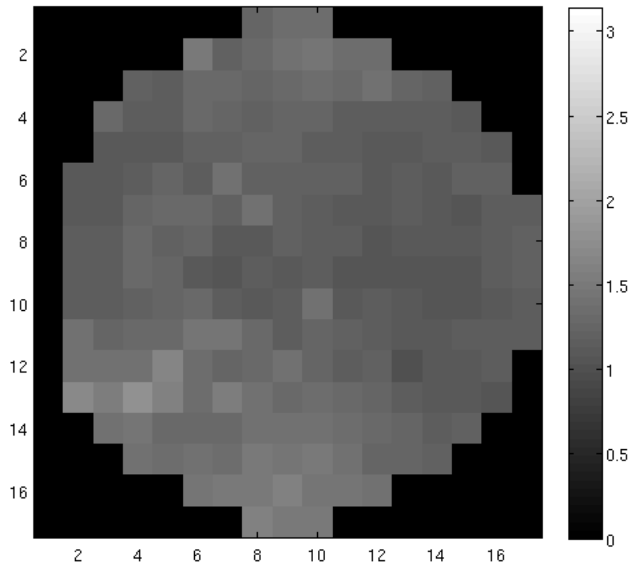

Figure B.8: Blank Walls - Gradient Descent

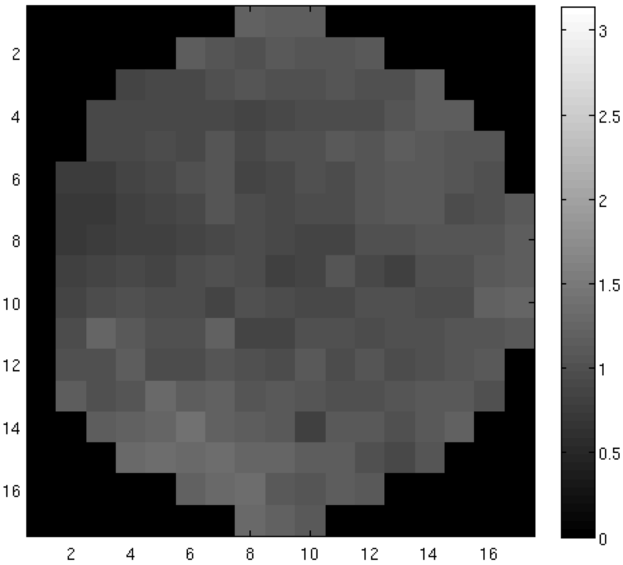

Figure B.9: Natural Scene - Gradient Descent

B.2 Rate of Return

B.2.1 Warping

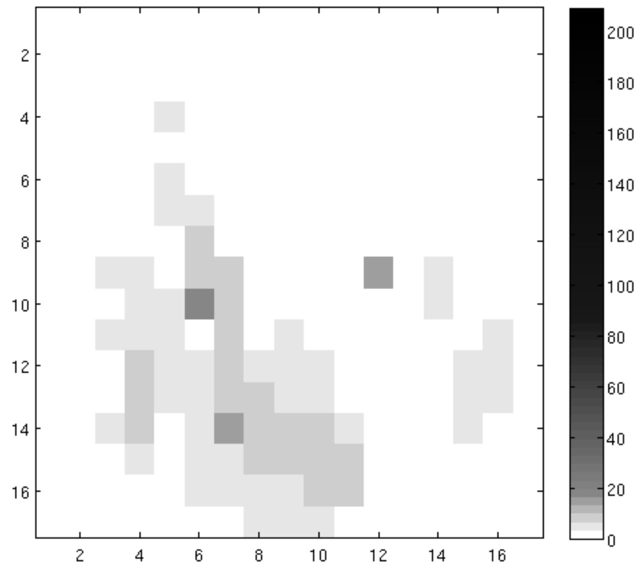

Figure B.10: 3 Landmarks - Warping

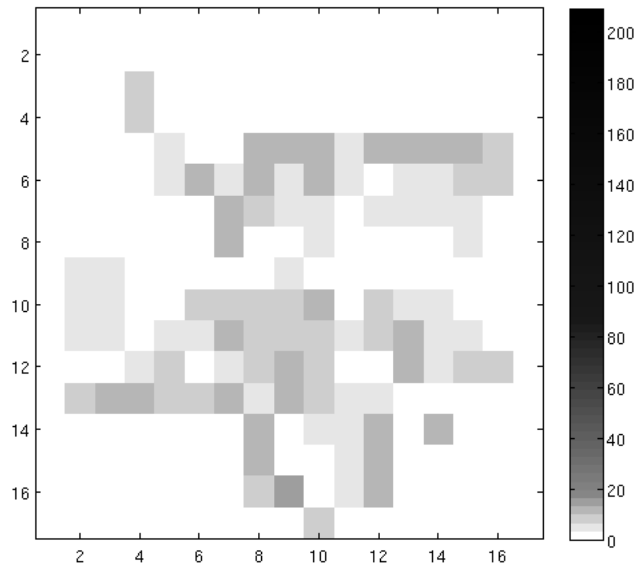

Figure B.11: Blank Walls - Warping

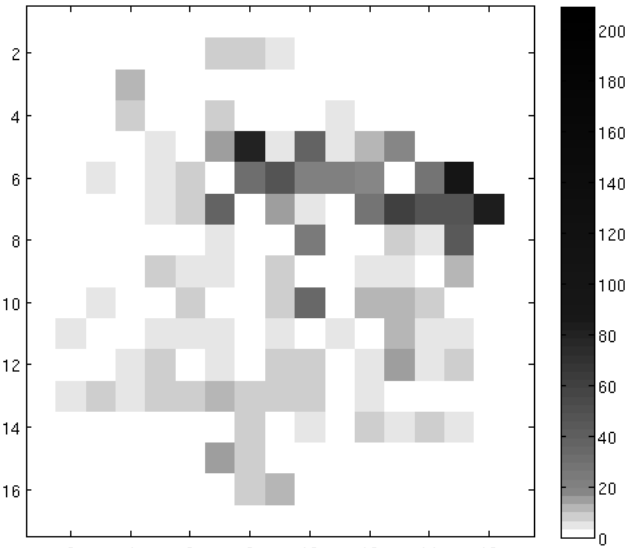

Figure B.12: Natural Scene - Warping

B.2.2 ALV

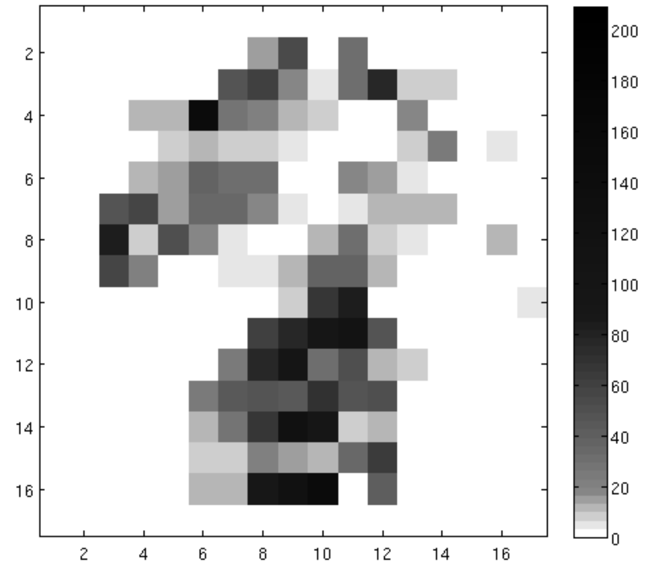

Figure B.13: 3 Landmarks - ALV

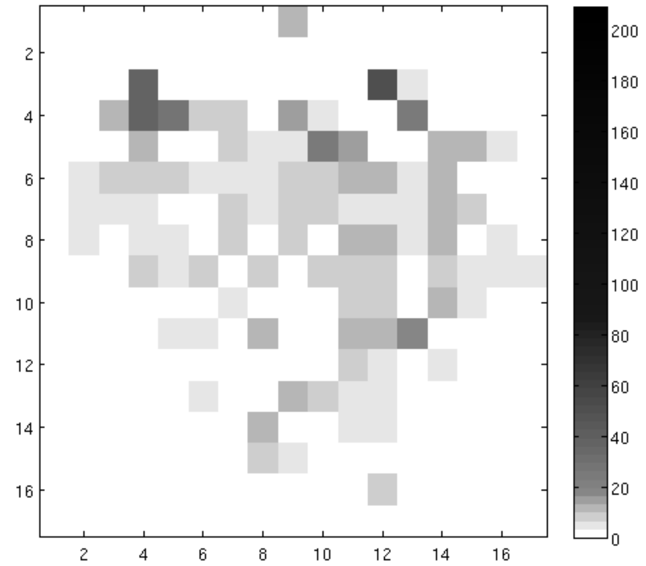

Figure B.14: Blank Walls - ALV

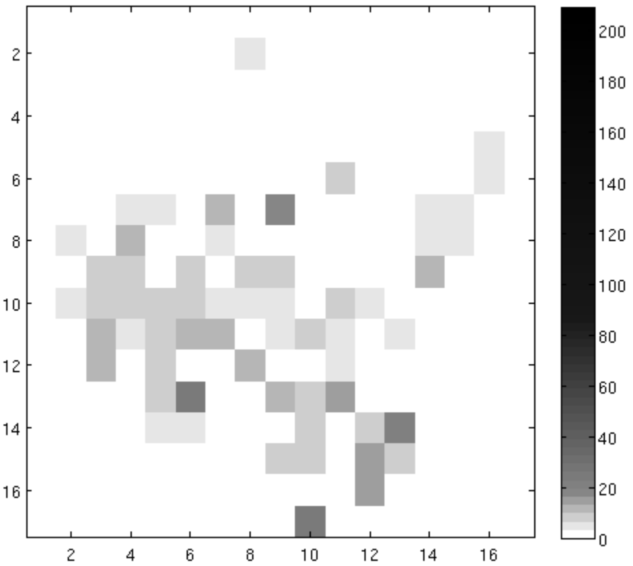

Figure B.15: Natural Scene - ALV

B.2.3 Gradient Descent

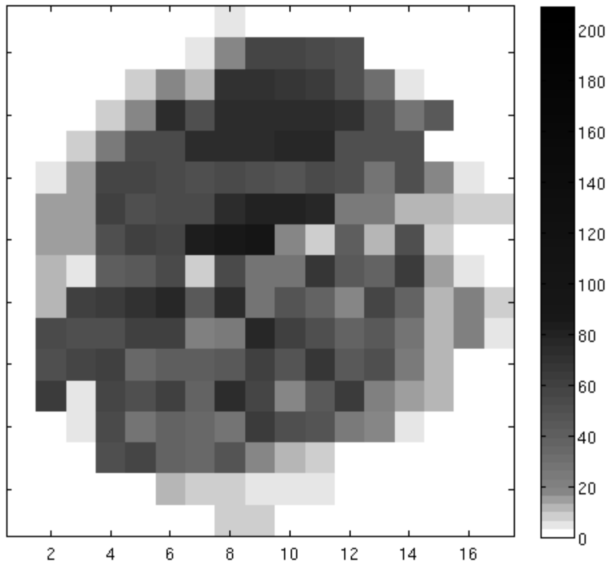

Figure B.16: 3 Landmarks - Gradient Descent

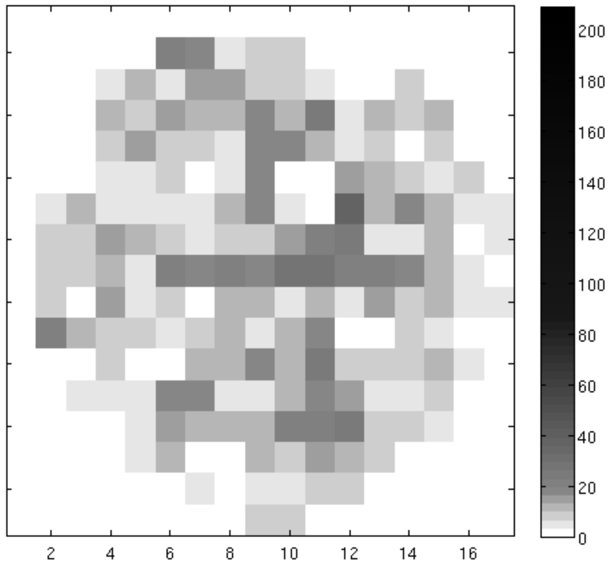

Figure B.17: Blank Walls - Gradient Descent

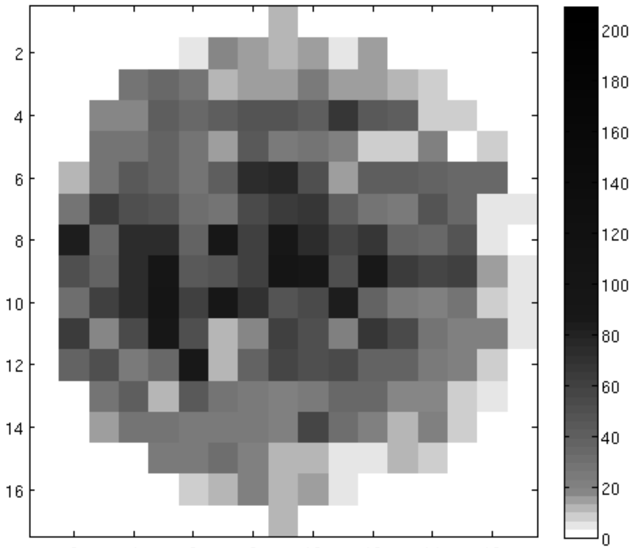

Figure B.18: Natural Scene - Gradient Descent

B.2.4 Gradient Descent (with previous locations)

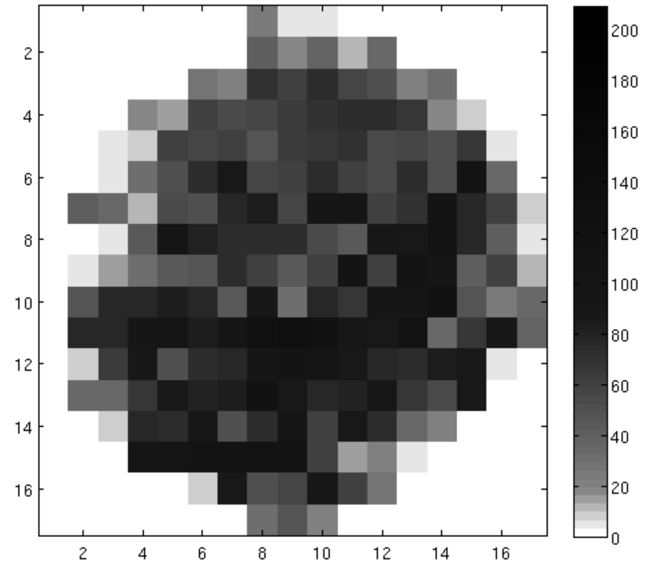

Figure B.19: 3 Landmarks - Gradient Descent (with previous locations)

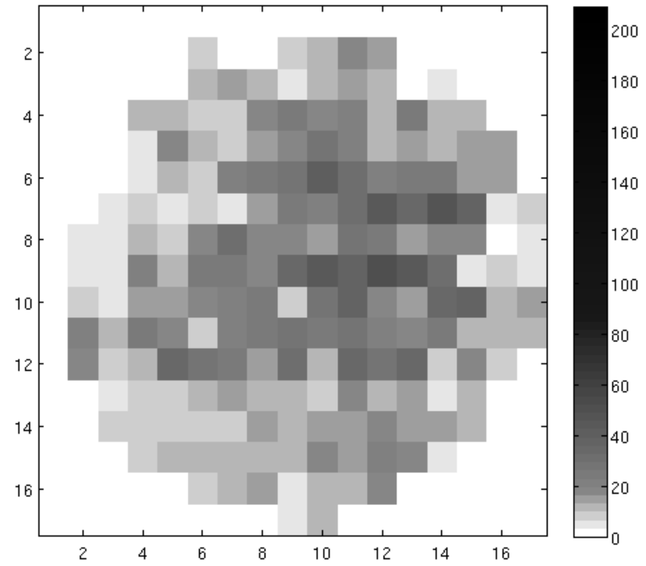

Figure B.20: Blank Walls - Gradient Descent (with previous locations)

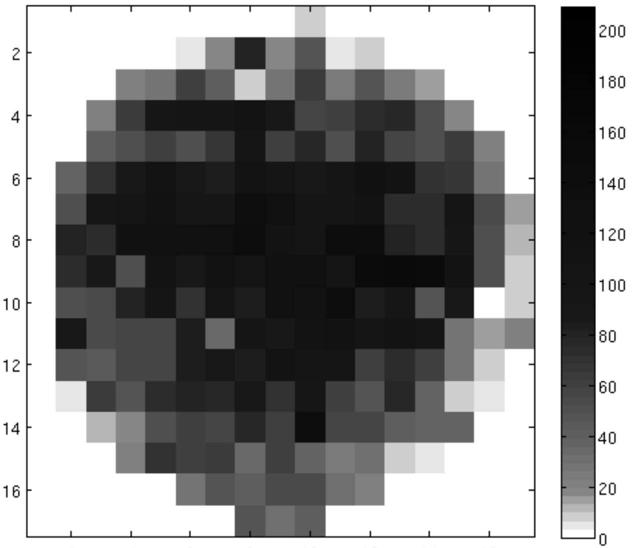

Figure B.21: Natural Scene - Gradient Descent (with previous locations)