Visual Homing for Micro Aerial Vehicles using Scene Familiarity

van Dalen, Gerald; Mcguire, Kimberly; de Croon, Guido

Publication date
2016

Document Version
Final published version

Published in
International Micro Air Vechicle Competition and Conference 2016

Visual Homing for Micro Aerial Vehicles using Scene Familiarity

Gerald J.J. van Dalen,* Kimberly N. McGuire and Guido C.H.E. de Croon
Delft University of Technology, The Netherlands

ABSTRACT

Autonomous navigation is a major challenge in the development of Micro Aerial Vehicles (MAVs). Especially when an algorithm has to be efficient, insect intelligence can be a source of inspiration. An elementary navigation task is homing, which means autonomously returning to the initial location. A promising approach uses learned visual familiarity of a route to determine reference headings during homing. In this paper an existing biological proof-of-concept is transferred to an algorithm for micro drones, using vision-in-the-loop experiments in indoor environments. An artificial neural network determines which control actions to take.

1 INTRODUCTION

A major challenge in robotics is to navigate autonomously through an unknown environment. Especially in indoor scenes, where no Global Positioning System (GPS) system is available, the entire navigation problem is not yet solved.

Current navigation algorithms either require expensive sensors or significant computation power. Especially Simultaneous Localization and Mapping (SLAM) methods have shown to be successful in real-time navigation, given enough computational power on-board a vehicle or good sensors. Most Micro Aerial Vehicles (MAVs) do not have such sensors and cannot perform heavy computations on-board the vehicle.

In order to find suitable navigation algorithms for MAVs, insects can be a source of inspiration, since they constantly have to deal with complex navigation problems while only having small-sized brains [1]. Different algorithms have already been created based on observations done on insects. A well-known example is using optic flow to get a sense of velocity, which is known to be done by insects [2]. Integrating this estimate for localization is called visual odometry. The obtained location estimate is employed in higher level navigation algorithms. Still, these algorithms are not readily available for tiny MAVs yet. One of the higher level skills employed by insects is the ability to return to the nest location. This is referred to as *homing* [3]. It would be an important enabler for MAVs, if they could use similarly high-level, but computationally efficient algorithms for navigation.

*Email address: gjj.vandalen@gmail.com

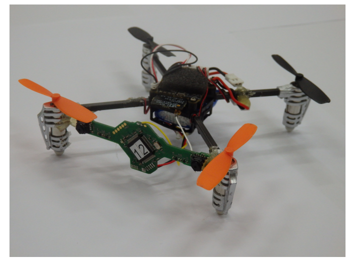


Figure 1: Pocket drone: a micro quad rotor containing a Lisa-S autopilot and a stereo camera [5]. While this pocket drone can already fly, stabilize and avoid obstacles, in this paper we investigate efficient insect-inspired algorithms that will allow it to navigate in an unknown environment.

A promising homing algorithm is proposed by Baddeley et al., where familiar views along a route are used to determine the correct direction to an earlier visited location [4]. This is a *visual* homing algorithm, since cameras are used as driving sensor. Instead of focussing on the contruction of a detailed (or coarse) map, Baddeley et al. propose that homing can be performed just by means of recognizing which direction seems most familiar to a robot. Furthermore, they use a small neural network to store and recapitulate a route in order to find the initial location. Potentially, this is very useful for MAV navigation algorithms, since it deals with limited storage capacity found on many small platforms, like the pocket drone shown in Figure 1.

In an effort to find efficient navigation algorithms for MAVs, this paper investigates the practical application of the scene familiarity algorithm on MAVs. The focus is on how robust familiarity is to determine control actions.

First, section 2 discusses the state-of-the-art in autonomous visual navigation on drones. Then, section 3 explains the scene familiarity method as introduced by Baddeley et al. Section 4 shows simulations and experiments for different environments, to overcome current shortcomings in the implementation described by Baddeley et al. Finally, closed-loop simulation flights are performed and presented in section 5, to show a more realistic use-case of view familiarity for MAV homing.

2 RELATED RESEARCH

This section gives a brief overview of previous research done to visual navigation and specifically visual homing. Visual SLAM is the most commonly used algorithm in camera-driven robotics. An example is shown in Motard et al., where an AIBO robot[1] must navigate back to its charging station [6]. Still, visual SLAM algorithms in real-time require much computational resources, since (visual) processing, mapping and self-localization must be performed simultaneously. Since most MAVs have limited computational resources, visual SLAM often cannot be run in real-time, which makes it less suitable for homing.

In 1983, Cartwright & Collett introduced the Snapshot Model [7]. The framework they presented gives an explanation of the navigation capabilities of bees when traveling between different food sources. The visual matching is done by a direct comparison of an image on the retina with a stored snapshot. The landmark approach is further extended by the addition of visual beacons [8]. A disadvantage of this is that many images have to be stored.

A similar approach uses Average Landmark Vectors (ALVs) to represent landmarks [9]. ALVs, introduced by Lambrinos et al. in 1998, are averages of the heading vectors to all landmark locations [10]. The homing vector is determined with respect to this ALV. ALV homing stores the location of interest as a vector, which is more efficient in computation and storage, than storing an entire image. However, due to its simplicity, ALV homing is also more prone to errors.

Scene familiarity methods refer to recognition of a traversed route, without specific information about the goal location. This means, a robot must always move into the *most familiar* direction. In the ideal case, this would automatically mean that the agent returns to the goal location. In 2012, a scene familiarity method is proposed for visual homing of desert ants [4]. The scene familiarity method proposed by Baddeley et al. is quite new and not yet used in robotic applications. The next section reviews their paper in depth.

3 THE SCENE FAMILIARITY METHOD

In an effort to find a biologically more plausible alternative to map-based navigation methods and the snapshot model described in the previous section, the scene familiarity homing method is introduced [4]. To show that homing navigation could take place without the use of visual odometry, a method is presented where views along the entire route determine the heading in which to proceed. Conceptually, this means that during a training run images in the direction of the route are stored. Then, when using the algorithm for homing, images taken around the robot are compared to these stored views in order to determine the most familiar direction.

When the homing capabilities are tested, the agent is placed back at its initial location. From there, homing is done by performing $360°$ scans of the world and comparing images taken in each direction with all images stored. A familiarity value of a single image is obtained by calculating the Sum of Squared Differences (SSD) of raw pixel values, as defined in Equation 1 [11].

[1]http://www.sony-aibo.co.uk/

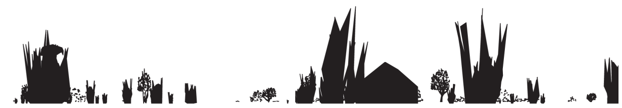


Figure 2: Binary panoramic image used in Baddeley et al. [4].

$$F(I) = -\arg\min_i \sum_{x,y} \left(I(x,y) - V_i(x,y)\right)^2 \qquad (1)$$

In this equation, $F(I)$ indicates the familiarity of view I, $I(x,y)$ is the current view and $V_i(x,y)$ are the stored views. It can be seen that the stored image that gives the closest match to the current image is used as familiarity value. The agent can rotate on the spot or use an omni-directional camera to obtain familiarity values in all directions. After determining the most familiar direction (by maximizing the values obtained with Equation 1), the simulated agent is moved in that direction.

The stored panoramas are binary images and have dimensions of 90 by 17 pixels (Figure 2). The resolution is such that each pixel in horizontal direction is equivalent to a rotation of $4°$. During homing, familiarity is evaluated for steps of 1 pixel, such that Equation 1 is evaluated 90 times. The maximum outcome of this results in the most familiar direction.

Due to the large memory needed for storing images and the computational requirements, the algorithm in the current form is not yet suitable for implementation on-board a small robot. Baddeley et al. therefore also study an unsupervised Infomax neural network to approximate familiarity [12]. The network is a two-layer neural network, where the linear combination of an input and the network weights represent familiarity. A lower value indicates *more familiar*. The training rule therefore adapts the weights such that the value is lower for every input encountered during training.

Baddeley et al. showed the validity of scene familarity with virtual robotic ants in a simulated environments. However, they use an environment of binary sceneries, which are not representative for the scenes through which a robot must navigate. Moreover, the simulation is set up such that moving the image by one pixel in the horizontal direction is equivalent to a rotation of the agent of $4°$. These direct relations to rotation and pixel difference are not realistic for real-life cameras. Furthermore, the algorithm has only been tested on relatively small distances, since images are stored every $4cm$ and movements of $10cm$ per timestep are made. When the method is implemented in robotics, the robot should be able to cover longer distances to make it more useful.

Recently, Gaffin et al. have published a detailed analysis on scene familiarity in realistic, indoor environments [13].

Distinguishing familiarity is both analyzed in rotation and translation, for raw pixel matching between images of different resolutions. A rail mounted camera is used to perform a MATLAB-driven experiment.

In our analysis of the scene familiarity method, we will use a simulator containing realistic sceneries, vehicle dynamics and camera parameters. A translation and rotation analysis will be performed as well, however, next to raw pixel values, we will also investigate alternative image representations, to determine which one is more suitable for recognizing familiar views. Closed-loop simulations with an MAV are presented and we show the use of an Infomax neural network as well, since this helps in meeting the limited storage requirements of an MAV. We hope to better understand autonomous navigation for small MAVs. In future work, the methods of Gaffin et al. should be applied to our simulation, to compare the obtained results.

4 Familiarity Analysis

In the previous section, the original simulation results presented by Baddeley et al. are discussed [4]. Based on this, a key question remains whether the algorithm will work in more realistic environments. In this and the following sections, an analysis of an indoor simulated environment is presented in combination with different image representation methods. First, the tested image representations and calculated performance measures are introduced. Then, simulation results of these different methods in multiple environments are shown. To validate this, similar results are shown on real imagery.

4.1 Methods

To test the usability of familiarity of scenes for visual homing, we investigate the familiarity sensitivity during both rotation and translation. Analyzing rotation is done by performing a 360° turn at a fixed location in the environment, in steps of 5°. A single image is stored and used as trained view and all other views experienced during this rotation are compared to this. The hypothesis is that familiarity should improve when the heading difference between the current view and the stored image decreases.

Translation is analyzed by evaluating familiarity in a grid of locations, with a fixed heading. Again, a single image is used as training sample and the familiarity is expected to improve when the distance to the trained view gets smaller. Results of this should show the sensitivity of familiarity with both increasing distance (in two directions) and increasing heading angle.

The following image representations are compared:

- **Raw pixel values** The sum of squared differences of each pixel in two images outputs a similarity score [11], as shown in Equation 1.
- **Texton histograms** Textons are small distinct image patches, which can be extracted from an image [14]. When clustered with a texton dictionary, histograms are formed which represent an image.
- **Hue Saturation Value (HSV) color histograms** Color histograms contain a classification of each pixel based on color intensity.

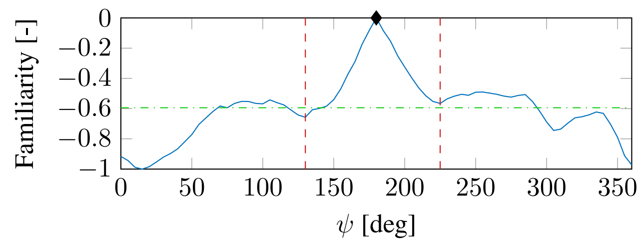


Figure 3: Rotation on the spot at a constant location in a simulator. Unfiltered images of 48 by 32 pixels are taken every 5° and compared to a stored image at a heading angle of 180°. The red dashed lines indicate the BoA bounds and the green dashed line shows the mean familiarity.

The performances of the different methods are evaluated by 1) looking at how distinct a view close to the trained view is, compared to other views and 2) what the probability is that the correct (i.e., trained) view is selected as most familiar, since that direction will be chosen for homing. Figure 3 shows an example of a familiarity evaluation when rotating on the spot. The trained image is positioned at an angle of 180° and, in this example, image matching is done using the SSD of raw pixel values. The performance is evaluated using the following measures:

- **Peak ratio** The peak ratio is defines as:

$$PR = \frac{\max F - \mu_F}{\max F - \min F} \tag{2}$$

  In this equation, F refers to the familiarity values shown in Figure 3 and μ_F is the mean of all familiarity values (i.e., the green line in the figure). The higher the peak ratio is, the more distinct a peak is.
- **Basin of Attraction (BoA)** The basin of attraction shows how far an agent can be off from the trained view, before diverging from the correct direction. It is evaluated by finding all local optima (both minima and maxima) and looking between which minima the agent converges towards the trained optimum familiarity (maximum).
- **Correlation coefficient** This is used to estimate the correlation between two neighboring heading angles, differing by 5°. Here, the Pearson product-moment correlation coefficient is used, where 1 indicates full positive correlation between two neighboring angles, -1 means full negative correlation and 0 means no correlation.

The BoA is considered to be most important, since it determines how far an agent can be off the route (i.e., the correct

(a) (b)

Figure 4: Examples from the scenery used in SmartUAV simulations (a) and the validation Cyberzoo environment (b).

heading), while still being able to converge back to the correct path with a gradient-like search. The peak ratio is mainly useful when an agent has no clue where to go; if the agent makes a 360° turn and the trained peak is very distinct, the probability of continuing in the right direction is high. The correlation coefficient gives a measure for how continuous a familiarity curve is. When the correlation is low, it could happen that spikes occur in the familiarity curve, which may give wrong results.

4.2 *SmartUAV Simulations*

This section shows analyses for sceneries in the SmartUAV simulator. SmartUAV is made for Guidance Navigation & Control (GNC) research on MAVs and specializes in the use of vision as primary sensor. The simulator is written in C++ and sensors and controllers can be connected using a block interface. This makes it easily extendable and the level of simulation fidelity can be adapted by changing complexity of vehicle dynamics, sensor dynamics and realism of the environment.

The tested environment is based on a sports hall located in Delft (the Netherlands). The dimensions are 30 by 60 meters. Figure 4a shows an example view of the sports hall. This environment is used for both familiarity analysis and closed-loop simulations.

As mentioned, both rotational and translational familiarity sensitivity will be tested. For familiarity estimation, SSD values of raw pixels, SSD values of texton histograms and SSD values of HSV color histograms are used and compared. The familiarity sensitivity to yaw rotations is most important for view familiarity-based homing. Each turn taken during homing is made based on the familiarity values for different heading angles. To analyze familiarity for different headings, different image representations are compared by calculating the BoAs, peak ratios and correlation coefficients. An MAV is simulated at a single location and stores a representation of one view. This view is matched to images in all other directions to get a measure of familiarity. This is done in a grid of locations in the sports hall, to get imagery in the center of the room, as well as close to walls. For each location, the BoA, peak ratio and correlation coefficient can be calculated.

Table 1 summarizes these performance measures for the different methods. The calculated BoAs, peak ratios and correlation coefficients are averaged for all locations and the standard deviations are included as well. Good performance is characterized by large BoAs (i.e., it is likely that the correct heading is found), large peak ratios (i.e., the correct familiarity value is distinct compared to familiarities in other directions) and correlation coefficients close to 1 (i.e., continuous and not too noisy).

	Raw pixels	**Textons**	**HSV**
BoA average	37.3%	36.7%	6.90%
BoA std. dev.	16.5%	12.0%	3.77%
Peak ratio average	0.57	0.43	0.53
Peak ratio std. dev.	0.10	0.076	0.13
Corr. coeff. average	0.98	0.98	0.80
Corr. coeff. std. dev.	0.051	0.0091	0.14

Table 1: Average performance metrics during rotation, for each image matching method in the simulated sports hall.

Looking at the results, it can be seen that the BoAs for raw pixel matching and texton histogram matching perform similarly. HSV histogram matching performs much worse, which is also seen in the lower correlation coefficient. This indicates more local optima, which inherently decreases the BoA. The peak ratio is best with raw pixel matching, although the differences between the different methods are quite small.

To illustrate the results shown in the table, familiarity curves are shown in Figure 5. The top plot shows raw pixel matching, the middle texton histogram matching and the bottom one HSV histogram matching. The blue, solid lines indicate the average familiarity curves for all locations in the environment, the red dashed lines indicate two times the standard deviation and the gray lines show some example familiarity curves at individual locations in the sports hall. The results are scaled such that the average lies between 0 and 1.

As expected, all average curves show a single peak at the trained locations (i.e., at 180°). The HSV histogram result however, shows a less predictable outcome, with a larger amount of local optima. This is in line with the lower BoAs and correlation coefficients shown in Table 1.

To test familiarity sensitivity with translation only, images taken in a grid pattern are analyzed. In the sports hall the trained view is obtained in the centre of the room, which is matched against views from the entire room, keeping the heading angle constant. In contrast to rotation, translational motion is not directly controlled. For homing, only the heading angle is adjusted in order to reach the correct destination. This means that good performance in translation is characterized by a familiarity that does not change too much for small displacements. Stated differently: when a 360° turn is performed, it is advantageous when the familiarity curves are similar for proximate locations, so that good homing performance is achieved even when exploration and homing routes do not perfectly align. Figure 6 shows the results in the sports

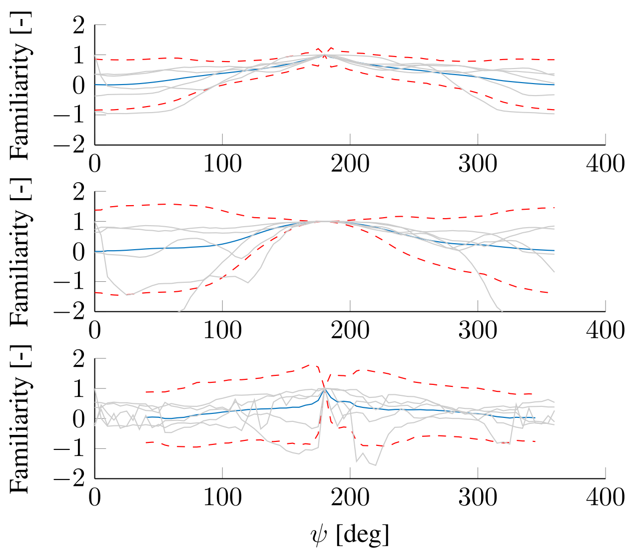


Figure 5: Average rotation on the spot of 231 locations in the sports hall environment in SmartUAV. Unfiltered images of 48 by 32 pixels are taken every $5°$ and compared to a stored image at a heading angle of $180°$. The red dashed lines indicate the 2σ bounds and the gray lines are some example familiarities. The top, middle and bottom plots indicate raw pixel matching, texton histogram matching and HSV histogram matching respectively.

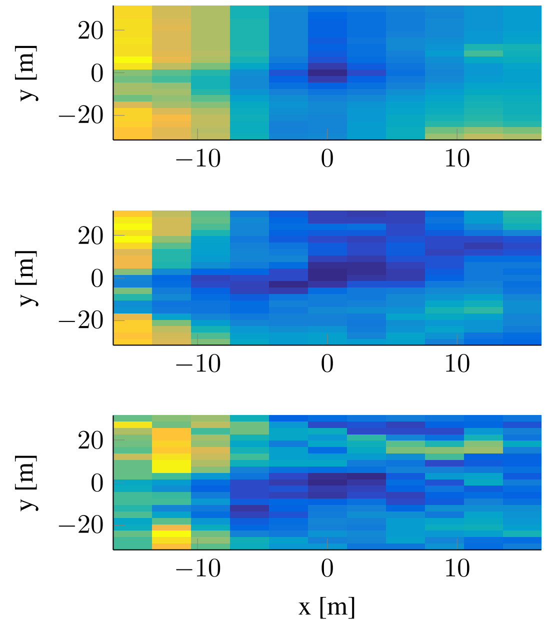


Figure 6: Varying x and y positions in a SmartUAV simulation in a sports hall, with constant heading angle. Unfiltered images of 48 by 32 pixels are taken in a grid pattern and compared to a stored image at the center of the grid (x=0 and y=0). The top figure uses raw pixel matching, the middle figure texton histograms and the bottom figure HSV histograms.

hall environment, for raw pixel matching, texton histogram matching and HSV color histogram matching. The colors indicate the familiarity of a certain location, where dark blue is most familiar.

From the figures it is clear that raw pixel matching shows the most distinct global optimum. Texton and HSV histogram matching however, show a larger region of optimal familiarity. This can be useful when the robot is slightly off-track, because rotational performance will be similar on different locations. However, both methods show several local minima, which can be disadvantageous for homing.

4.3 *Validation Experiment*

The previous analysis is done in simulation. To validate this, an experiment is shown using real imagery taken in an indoor environment. The environment used is the Cyberzoo; a flight arena located at the TU Delft, as shown in Figure 4b.

Validation is done for both rotation and translation. For rotation, videos of rotations on the spot are recorded, containing 25 videos in a grid of 5 by 5 meters. The average BoAs, peak ratios and correlation coefficients are computed, as in the simulations presented in the previous section. The results, including the corresponding standard deviations, are shown in Table 2. The first observation is that the BoAs are much smaller than in simulation. This is explained by more spikes (and hence local optima) in the results, which is confirmed by the lower correlation coefficients. It is however, in contrast with the observation in the previous section that more realistic environments yield higher BoAs.

The second observation is that texton and HSV histogram matching show slightly better BoAs than raw pixel matching. Due to the small differences and the large standard deviations however, no significant conclusions can be drawn from this. The corresponding rotation plots are shown in Figure 7.

	Raw pixels	**Textons**	**HSV**
BoA average	9.13%	12.7%	11.7%
BoA std. dev.	3.38%	6.57%	4.24%
Peak Ratio average	0.53	0.41	0.37
Peak Ratio std. dev.	0.054	0.095	0.093
Corr. Coeff. average	0.82	0.92	0.84
Corr. Coeff. std. dev.	0.093	0.025	0.14

Table 2: Familiarity performance metrics for each image matching method in the Cyberzoo environment.

Translation is validated by comparing images taken facing the same direction, in a grid of 49 locations. The results are quite similar to the simulation results and are shown in Figure 8. Again, the result for raw pixel matching shows a very narrow peak at the trained location. This can be disadvantageous for homing, since a small offset from the training

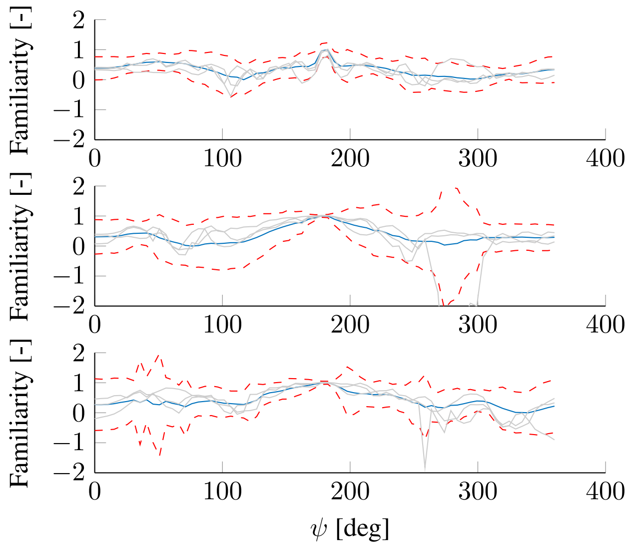


Figure 7: Average rotation on the spot of 25 locations in the Cyberzoo environment. Unfiltered images of 64 by 36 pixels are taken every $5°$ and compared to a stored image at a heading angle of $180°$. The red dashed lines indicate the 2σ bounds and the gray lines are some example familiarities. The three plots indicate raw pixel matching, texton histogram matching and HSV histogram matching respectively.

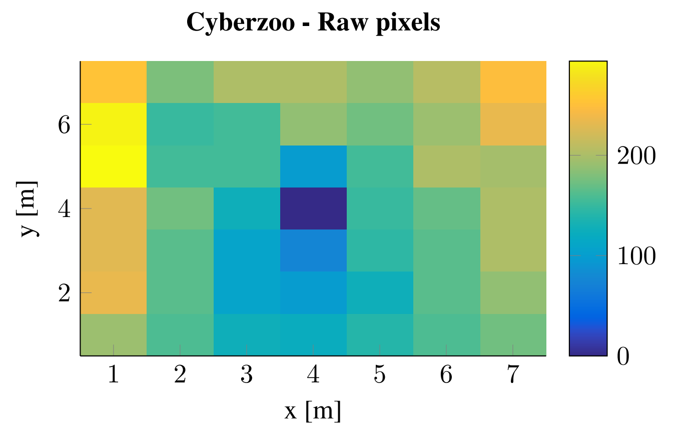
path can cause divergence from this path. When looking at the texton histogram matching result, it can be seen that two clear optima are present. Even though the surrounding region has quite similar familiarity values, the local optimum at $x = 3$ and $y = 2$ might result in wrong convergence.

Looking at both rotation and translation of HSV histogram matching, it can be observed that the real-life results are better than those made in simulation. This can be explained by more distinct colors in the validation imagery, such that more bins in the HSV histogram are filled.

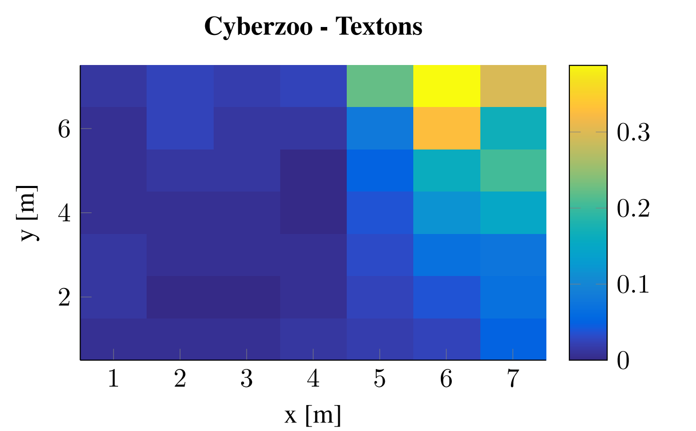
5 Closed-loop Simulation Flight

As mentioned in the previous sections, the recognition of views during rotation performs best for both raw pixel matching and texton histogram matching. Especially in simulation, the BoAs of these two methods are comparable. When observing familiarity during translations, both texton and HSV histogram matching show a large central region of similar familiarity. As explained earlier, this can be advantageous for homing, since recognizing the correct heading during rotations probably yields the same result for proximate locations. When looking at closed-loop results it is therefore expected that texton histogram matching will perform better than the other two methods.

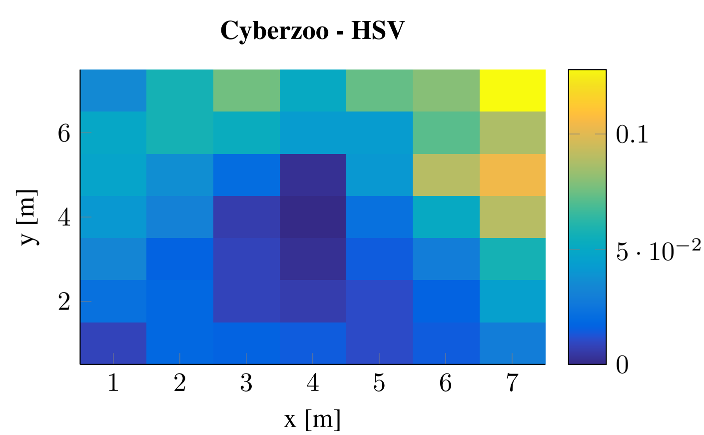
To show a closed-loop simulation, a simulated robot is placed in the sports hall environment. A route is learned by flying backwards (with a speed of $0.5 m/s$), such that the front



(a)



(b)



(c)

Figure 8: Varying x and y positions using pictures of the Cyberzoo environment, with constant heading angle. Unfiltered images of 64 by 48 pixels are taken in a grid pattern and compared to a stored image at the center of the grid (x=4 and y=4).

camera looks in the homing direction, which is necessary to use scene familiarity for homing. One third of the image taken at the center is used for training. When homing is initiated, the robot starts flying forward with a constant speed of $0.5m/s$ and the heading is constantly determined using view familiarity. This is done by selecting one third of the image giving the best match with one of the trained views. The center of this image patch is converted to an angle, to which the MAV is steered. Views are obtained from a forward looking camera, with a field of view of $90°$. The result is shown in the left part of Figure 9. Here, the blue solid line is the training route, starting at $x = 4m$ and $y = 12m$, which are arbitrarily chosen. A route of approximately $20m$ is flown.

From the results it can be seen that both texton histogram matching and HSV histogram matching approximately reach the initial location. The main difference is that texton histogram matching performs turns with a small delay, where HSV histogram matching turns too early. The delay can be explained by low frequency: because all possible patches are extracted from each image, texton histogram matching operates at approximately $1Hz$, where HSV histogram matching operates at approximately $20Hz$. Texton histogram matching can be significantly improved by using sub-sampling of textons, instead of extracting all. For HSV histogram matching it could be questioned whether it only performs well because the flying direction is approximately straight. When homing is done by matching raw pixels (performed at approximately $5Hz$), the robot diverges from the trained route. It does, however, follow the curvature of the trained path. The fact that raw pixel matching works worst suggests that differences in familiarity when performing small translational movements causes views to be hard to recognize.

As mentioned, the Infomax neural network can be used as function approximator of familiarity [12]. To test this in closed-loop, the three methods are all represented in a neural network. For both texton and HSV histogram matching a network with 50 inputs is defined (i.e., each histogram forms one input vector to the network). The number of novelty neurons is arbitrarily chosen to be 200. Furthermore, the number of epochs is set to 500. It turned out that a lower number of epochs gives significantly worse performance. In further simulations or flight tests this should be tuned by testing multiple numbers of both novelty neurons and epochs. For raw pixel matching, the image is scaled down to 16 by 12 pixels, which gives 192 inputs to the network. Larger dimensions as input cannot be performed in real-time. The number of novelty neurons and epochs are kept the same.

The results using an Infomax network can be seen in the right part of Figure 9. It is clear that the results are slightly worse than with a perfect memory (i.e., by keeping a database of images, texton histograms or HSV histograms). It does however, look quite similar to the perfect memory case, which suggests that the assumption that Infomax is only used as approximator for views is quite good.

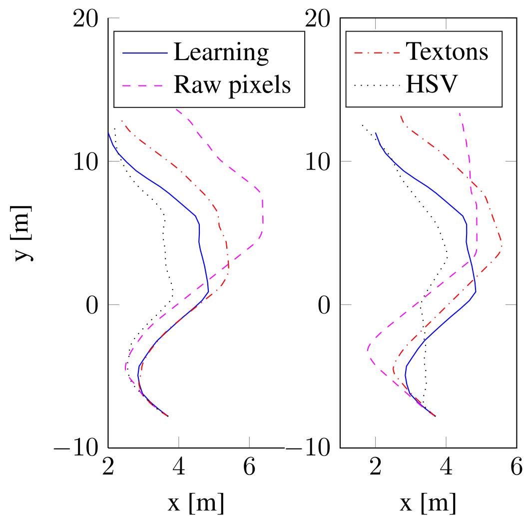


Figure 9: Closed-loop homing simulation in the sports hall environment in SmartUAV. On the left, a perfect memory is used; on the right the Infomax neural network is applied.

6 Discussion

When first looking at the rotational analysis, it was observed that raw pixel and texton histogram matching performed best. When looking at the translation results, raw pixel matching shows the most distinct peak. Because position of the robot is not directly controlled, it is advantageous that a large familiar region appears in translation, so that a small displacement of the robot does not change the homing performance. This was especially the case for texton histogram matching and HSV histogram matching. This suggests that texton histogram matching would perform best, which is confirmed by the closed-loop results. Surprisingly, HSV histogram matching shows very good performance in closed-loop. A reason for this can be that generating and storing HSV histograms is computationally very efficient, which allows for a low timestep. This means corrections are made very quickly so that the robot does not diverge too much. It does not say however, that HSV histogram matching would perform well when divergence already happened.

When evaluating the closed-loop tests in this paper, some limitations can be identified. First of all, it is only tested in simulation. Although the fidelity of the simulation is higher than the simulations performed by Baddeley et al., it is questionable whether the same results would be obtained in a real flight. Furthermore, additions can be proposed to make the algorithm more robust. An example is to use active rotation instead of using the inherent field of view of the forward looking camera, such that bigger turns can be made. Alternatively, a camera with a larger field of view can be added. Another possibility is the use of visual odometry to get a rough estimate of the path taken. Odometry could be used to prevent severe divergence from the correct route. Since the experiment enforces small turns only, it cannot yet be concluded that the method works well for diverse trajectories.

Another point of discussion is that the main reason scene familiarity can be a viable approach for visual homing of MAVs is computational efficiency. The only way this is tested in this paper, is by performing closed-loop real-time simulations on a laptop computer. When implementing the algorithm on-board an MAV, the real-time performance may be inadequate due to a slower micro-processor. The one exception was HSV histogram matching, because both the computations needed to extract histograms, and the storage capacity are limited. In this paper however, all textons were extracted from each image. Usually, it suffices to randomly pick a set of textons, which would drastically improve computational performance. The storage of a texton histogram is similar to storing an HSV histogram. A huge advantage of using a neural network is that the storage capacity is constrained. Even though this means that the network can *forget* earlier trained views (which is also investigated by Baddeley et al.), it allows control over the often very limited storage capacity on MAVs. Training on the other hand, is quite slow; especially when having to train each sample 500 times.

7 Conclusion and Recommendations

This paper investigates the applicability of the scene familiarity homing method, observed from insect behavior, to MAVs. The scene familiarity method is introduced as proof of concept for desert ants to use the recognition along a route to find their way home. Next to this, an unsupervised neural network was used to keep storage of familiarity compact.

The concept of only using recognition along a route is a very interesting one. The analysis shows the closed-loop performance is good. The reason the method is promising, is the computational efficiency. Especially HSV histogram matching showed surprisingly good closed-loop performance while running quite fast. For the other two image representations the algorithm works in real-time on a laptop, although the frequencies in the current implementations are low.

It is concluded that using texton or HSV histogram matching is useful for visual homing on small robots. Once a route is lost, the risk of divergence is quite high. This must be further investigated. It seems very useful to combine scene recognition with existing methods like visual odometry. Especially when some more thought is put in optimizing the algorithm and sensor usage (like multiple or omnidirectional cameras), two computationally efficient methods can be combined to succesfully perform homing.

References

[1] Z. Mathews, M. Lechon, J.M. Blanco Calvo, A. Dhir, A. Duff, S Bermudez i Badia, and P.F.M.J. Verschure. Insect-like mapless navigation based on head direction cells and contextual learning using chemo-visual sensors. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, New York, USA, oct 2009. IEEE.

[2] M.V. Srinivasan. Where paths meet and cross: navigation by path integration in the desert ant and the honeybee. *J Comp Physiol A*, 201(6):533–546, may 2015.

[3] R. C. Nelson. Visual homing using an associative memory. *Biological Cybernetics*, 65(4):281–291, aug 1991.

[4] B. Baddeley, P. Graham, P. Husbands, and A. Philippides. A model of ant route navigation driven by scene familiarity. *PLoS Computational Biology*, 8(1), jan 2012.

[5] B.D.W. Remes, P. Esden-Tempski, F. Van Tienen, E. Smeur, C. De Wagter, and G.C.H.E. De Croon. Lisa-s 2.8g autopilot for gps-based flight of mavs. Delft University of Technology, 2014.

[6] E. Motard, B. Raducanu, V. Cadenat, and J. Vitria. Incremental on-line topological map learning for a visual homing application. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, New York, USA, apr 2007. IEEE.

[7] B. A. Cartwright and T. S. Collett. Landmark learning in bees. *Journal of Comparative Physiology*, 151(4):521–543, 1983.

[8] T. S. Collett. Insect navigation en route to the goal: Multiple strategies for the use of landmarks. *Journal of Experimental Biology*, 199:227–235, 1996.

[9] D. Lambrinos, R. Möller, T. Labhart, R. Pfeifer, and R. Wehner. A mobile robot employing insect strategies for navigation. *Robotics and Autonomous Systems*, 30(1-2):39–64, jan 2000.

[10] D. Lambrinos, R. Möller, R. Pfeifer, and R. Wehner. Landmark navigation without snapshots: the average landmark vector model. In *Proceedings of Neurobiology Conference Göttingen*, 1998.

[11] J. Zeil, M.I. Hofmann, and J.S. Chahl. Catchment areas of panoramic snapshots in outdoor scenes. *Journal of the Optical Society of America A*, 20(3):450, mar 2003.

[12] A.J. Bell and T.J. Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural Computation*, 7(6):1129–1159, nov 1995.

[13] D.D. Gaffin and B.P. Brayfield. Autonomous visual navigation of an indoor environment using a parsimonious, insect inspired familiarity algorithm. *PLOS ONE*, 11(4):e0153706, apr 2016.

[14] M. Varma and A. Zisserman. A statistical approach to texture classification from single images. *Int J Comput Vision*, 62(1-2):61–81, apr 2005.