Appearance Based Processes for Visual Navigation

Stephen D. Jones, Claus Andresen and James L. Crowley
Project PRIMA-IMAG
Institut National Polytechnique de Grenoble
46 Ave Félix Viallet
38031 Grenoble, France

Abstract[1]

This paper describes the use of appearance based vision for defining visual processes for navigation. A visual processes which transform images to commands and events. A family of visual processes are defined by associating the appearance of a scene from a given viewpoint with the simple trajectories. Appearance is captured as a set of low-resolution images. Energy normalised cross correlation is used to maintain heading, to estimated confidence and to servo control a robot vehicle while following a path. Experimental results are presented which compare results with a single camera, a pair of parallel cameras and a pair of divergent cameras. The most accurate (and robust) navigation is found with a pair of cameras which are slightly divergent.

1. Introduction

Visual navigation requires fast and robust image processing. Traditional reconstruction approaches have not provided an adequate solution for navigation because of the time taken for image processing, and the unstable nature of such processing. Image segmentation, line segment extraction and stereo correspondence are time-consuming processes particularly when additional effort is require to stabilise the resulting image primitives. Reconstruction techniques often rely on a precise camera calibration and known operating parameters. Cameras mounted on moving robots require specialised engineering to approach the level of precision needed.

Appearance based vision techniques [Pentland-Turk 91] provides an alternative approach for the use of computer vision for navigation. Such techniques provide a means to define simple processes which transform images into commands for displacement and steering and corrections to estimated position and orientation. Such techniques are also easily used to design simple visual processes which transform images to information for navigation.

This paper concerns the use of appearance based techniques for a number of simple navigation tasks. We describe the use of appearance based methods to design simple visual processes for navigation. We present the results of an experiment which is designed to assess the effectiveness of an appearance based approach to navigation, in single and multiple camera configurations on a distributed system.

1.1 The Task

Our basic experimental task is to navigate from a home position to an office in the ground floor of our building. The office is on the same floor and within the same building. Navigation includes crossing an open room turning to face a doorway, traversing the doorway and negotiating a narrow corridor. The robot senses the world using cameras and odometry. The environment is structured and stable but not static. The robot must stop when its way is blocked, or when confidence in sensor processing drops below a predefined minimum threshold.

1.2. Integration of Visual Processes

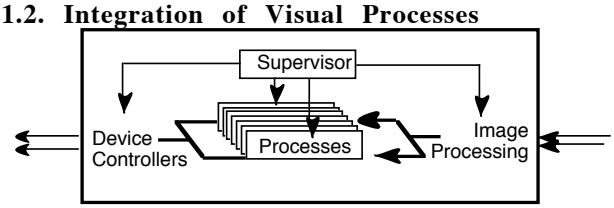

Figure 1. The Architecture for Integrating multiple visual processes

Our system is based on the SERVP (Synchronous Ensemble of Reactive Visual Processes) architecture for integrating visual processes [Crowley-Bedrun94], shown in figure 1.. In this architecture, visual processes are defined as transformations from images to parameter vectors or events. The architecture operates as a cycle in which a supervisor schedules and activates a small subset of the available processes. The set of processes are selected based on system goals and events.

Our experiments are preformed with an implementation of this architecture (the RAVI System, [Zoppis 97], using an interpreter equipped with dynamic linking and distributed processing under PVM. In this system, a visual processes is represented as a transformation from image to measurement or image to event.

Our implementation of supports asynchronous distributed processing in a hierarchy. The asynchronous distributed approach in turn supports true parallel processing; the images from multiple cameras are processed independently and the results later fused. The speed gains from parallel processing in turn advance the

[1]Funded by the CEC DG II - Programme TMR

possibility of using close-coupled active vision for control.

The task hierarchy in our system supports the purposive planning and incremental refinement that selects and schedules processes for application, in this way constraining processing in support of the task in hand. In this experiment we are able to use the architecture and to apply it to the control of visual processes for navigation.

2. Appearance Based Navigation

Appearance based approaches to vision have proven to be robust [Schiele96] [Turk-Pentland 91]. in this experiment we have attempted to exploit this property with the application of an appearance based technique to navigation. The work builds on the "View Sequenced Route Representation" (VSRR) [Matsumoto96a] approach and advances the work in terms of camera configuration and control of processing.

Figure 2a Images from database of Laboratory.

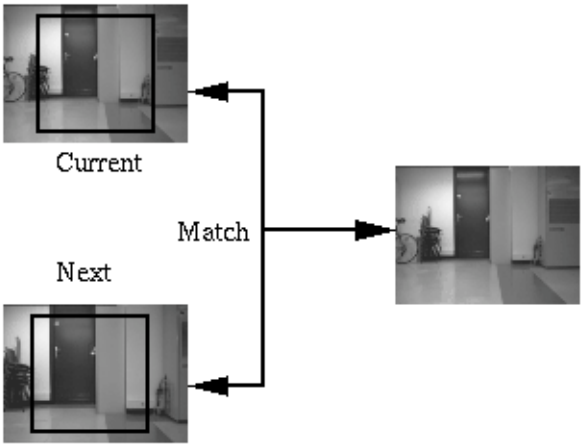

Figure 2b. Correlating on sequential images

In our experiments, we use zero mean energy normalised cross correlation (ZNCC, see [Crowley-Martin 95]) applied match an MxN region R(m, n) of the observed image, and to a template image, T(m, n). The template image and each neighborhood (i, j) of the observed image are normalised by subtracting the mean.

$$\mu_T = \frac{1}{MN} \sum_{m=0}^{M} \sum_{n-0}^{N} T(m, n)$$

$$E_T^2 = \sum_{m=0}^{M} \sum_{n=0}^{N} (T(m, n) - \mu_T)^2 \;) \;)$$

$$\bar{T}(i, j) = \frac{T(i, j) - \mu_T}{\sqrt{E_T^2}}$$

$$\mu_R(i, j) = \frac{1}{MN} \sum_{m=0}^{M} \sum_{n-0}^{N} R(i+m, j+n)$$

$$E_R(i, j)^2 = \sum_{m=0}^{M} \sum_{n=0}^{N} (R(i+m, j+n) - \mu_R(i, j))^2)$$

$$\bar{R}(i, j) = \frac{R(i, j) - \mu_R(i, j)}{\sqrt{E_R(i, j)^2}}$$

$$ZNCC(i, j) = \sum_{m=0}^{M} \sum_{n=0}^{N} (\bar{R}(i+m, j+n)\, \bar{T}(i, j) \quad (1)$$

Navigation with this approach relies on establishing a network of view-points to which are associated a set of images of the environment. When navigating, these images are correlated with live-video signals and used to control the movement of the robot and to correct the estimated position maintained by odometry.

To collect images, the robot is driven forward along a desired trajectory, and low resolution, wide angle images are acquired. The first image along a trajectory is taken as a reference image and stored. Subsequent images are compared to this first image using ZNCC (Eqn. 1). When the correlation peak drops below a pre-established threshold, the image is taken as the new reference image and saved along with the current estimated position.

When navigating autonomously, ZNCC is used to correlate the live images with those in the database. A correlation peak is established and translated into a control signal to drive the robot. The database is traversed sequentially as the robot follows the trajectory; correlating only with the current and the next image in the database and advancing through the database as correlation with the next image is confirmed to improve on that of the current image.

Actions are associated with images, using the correlation value as a confidence measure for the proposed action. Confidence below a given threshold indicates an unacceptable level of uncertainty in the action to be taken and leads to the default action of stopping the robot. A low correlation may indicate that the robot has strayed from the target trajectory or that the current conditions

differ from those under which the images were collected, this may be the result of obstacles blocking the way or substantial changes in lighting.

2.1. Perception -> Action

Associated with the current image is an action, usually to drive the robot in such a way that live images correlate well with stored images. Other actions may however be associated with images, rotating the robot in order to make a right-angled turn or stopping the robot, for example when the current image includes a closed door and the trajectory of the robot is blocked. Perception may be limited to the image taken by a single camera or incorporate the images from two cameras, include sonar or other perceptual data, together reinforcing the choice of an appropriate action.

The technique was originally developed for a single static camera mounted with its optical axis perpendicular to the drive axis. In this configuration the position of the correlation peak is used to control only the steering angle of the robot. The offset of the peak from the centre indicating the need for the robot to turn to the left or to the right.

The technique was subsequently extended to using a pair of steerable cameras. With two cameras, we need to combine the information from two correlation peaks. If the two cameras are positioned with their optical axes perpendicular to the drive axis, they offer additional robustness in terms of redundancy. Each correlation peak should approximate to the same offset. Control may then be effected with a weighted average, or other fusion scheme, based on the strength of the correlation peak. Two cameras in a parallel configuration offer robustness but again control only the steering angle.

Setting the two cameras with an angular offset introduces additional complexity and access to another degree of freedom. However, the offset angle allows a form of triangulation providing some knowledge of the robot's position along the trajectory as well as its heading. If we assume that the cameras are offset in a symmetric configuration about the normal to the drive axis we can combine the correlation results to control both steering angle and velocity. Steering angle is determined as in the simpler parallel case, but now we may use the combined correlation peak offsets to determine position and so control velocity.

2.2. Experimental Implementation

Our system controller combines five distributed compound processes running on multiple machines (under PVM) with a number of component processes. ABC-2 acts as a centralised supervisor combining the results of the two ABC-1 correlation processes and commanding the robot via the Relay process. A fifth process the RoleServer (not shown) provides a naming service . The two ABC-1 processes are each constrained to run on specified machines in order to access directly the images acquired at runtime. Relay is similarly limited to the host that maintains the radio link with the robot. All other processes are distributed over the network at runtime.

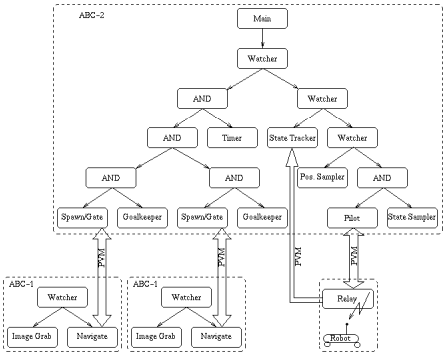

Figure 3. The hierarchy of processes used to test appearance based navigation.

The Process **ABC-2** is responsible for spawning the two image processes and commits components to maintaining a communication interface, in order to receive data from and send commands down to the subprocess as required. Throughout the structure of ABC-2 we repeatedly see the pattern of the left branch of a composition supplies data to the right branch which processes the data in order to provide control events.

The Process **ABC-1** is a simple composition of component processes that acquires images, correlates them with a data-base and passes the results up to the parent.

The Process **Relay** is a way station for communication with the robot. Only processes running on a given host may communicate with the robot via the serial radio link. Relay provides a flexible way to manage this processing bottle-neck by allowing remote processes to communicate with the robot via a gateway.

2.3. Processes

The component processes within ABC-2 and ABC-1 perform both specific and generic functions; tasks specific to the particular application as well as more general purpose tasks that may equally be found in other robotic applications. The following is a description of these processes.

Goalkeeper detects the arrival of the robot at a goal point. The robot arrives at the goal point when the correlation peak of the goal image is at a maximum. Initially, the maximum was considered to have been detected when there was a drop in correlation of 10%, this was later refined with the detection of a negative gradient on the tangent to the curve of maximum correlation over time. Given sufficient data an SPC solution would have been a worthwhile consideration.

ImageGrabber grabs the image from the image acquisition board and down-samples it to the size used in correlation. Down-sampling is not done onboard the

acquisition card as it was found to introduce aliasing and other unwanted image processing side-effects.

Navigate takes the down-sampled images provided by an ImageGrabber and correlates them with the images in the database. The quality and position of the correlation peak along with a proposed corrective steering angle is then exported to the parent gateway process.

PositionSampler constantly queries the robot for its estimated position. In these experiments the position data is not used in the control of the robot, but as an aid to monitoring for performance comparison. We are able to compare the robot's estimated position with that resulting from visual-servoing.

Pilot is the fusion process, concerned with combining the results of correlation as they become available and converting those results into steering commands to be issued to the robot.

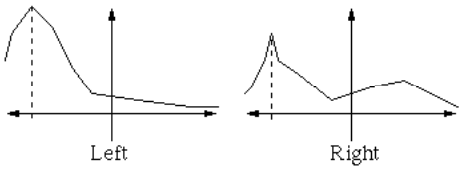

Figure 4a: Peak-pattern indicating turn left

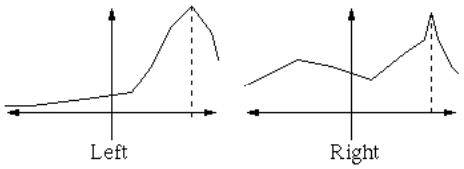

Figure 4b: Peak-pattern indicating turn right

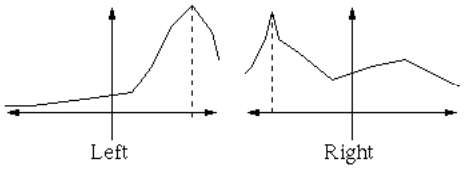

Figure 4c: Peak-pattern indicating to move ahead

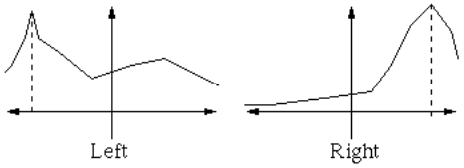

Figure 4d: Peak-pattern indicating move back

With a single camera, when the correlation peak is left of centre for the current live image, the robot is turned to the left, and there is a symmetric relation for the peak right of centre. The peak must be offset a certain distance from the centre before control is initiated, the confidence in the steering command corresponds to the confidence in the correlation. Steering was calibrated by assuming an image plane parallel to the image contents.

When we have a pair of cameras, interpretation is a little more complicated. There are now four possibilities, shown in figure 4. The system provides a confidence measure that is the combination of two results and may also detect inconsistencies in results. We expect the offset from the two peaks to approximate to the same absolute value, when this is not the case, the robot is in a position inconsistent with that used to take the original database images. In these experiments, we restricted ourselves to control of the steering angle in order to make a direct comparison with the single camera approach.

Spawn/Gate is the process that spawns the image acquisition and correlation sub-processes. Once a sub-process is established, the parent component becomes a gateway for the data that it provides, converting messages sent over the PVM network into data exported to local component processes via ports.

StateTracker maintains a copy of the robot state, this ensures that for example steering commands are not sent to the robot when it is rotating at a sub-goal point. In general the state of the robot is maintained so as to avoid issuing commands that will conflict with its current state.

Timer is used to record the current time, so that trace and log data can be recorded and indexed relative to the clock running on each host. Host clocks are periodically synchronised by operating system demons.

3. Performance Evaluation

In order to test the accuracy and robustness of a number of navigation strategies a number of tests were performed. Ultimately the research has the task of providing the robot with the capacity to navigate from the home position to an office but for the purposes of a control experiment a simpler sub-task was selected.

The robot is to navigate around the laboratory following a rectangular circuit approximately 9m by 3m. The robot moves in a straight line between corners, making a 90° turn in each corner. While the robot moves we are able to record its position according to odometry and at each corner to measure its actual position in the real-world. In this way we can calculate an error and assess the accuracy and stability of the various techniques.

The circuit does not exhaustively test the capabilities of all of the techniques, rather it tests their suitability for a given task while highlighting some strengths and some weaknesses. The approaches are diverse in nature and naturally each will be suited to particular environments. The modular aspect of the process framework supports integration and provides the common platform for testing and integration. For these experiments when testing the single camera configuration we were able to switch off one camera and use the application exactly as developed for a pair of cameras.

4. Experimental Results

The histograms below (Fig. 5) summarise some of the

results collected, showing the evolution of error in the position of the robot as it arrived at each corner (vertex) in the test trajectory. Some open and close loop odometry controlled experiments are included for comparison purposes.

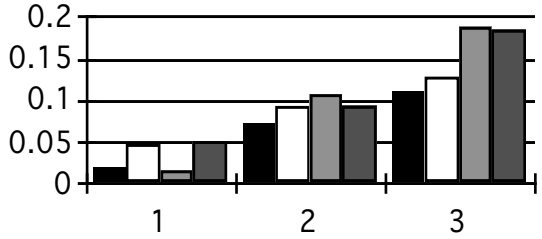

Figure 5.a Position Error (in meters) at each corner for three consecutive laps around test course when using open loop odometry. Error is measured at corners of circuit.

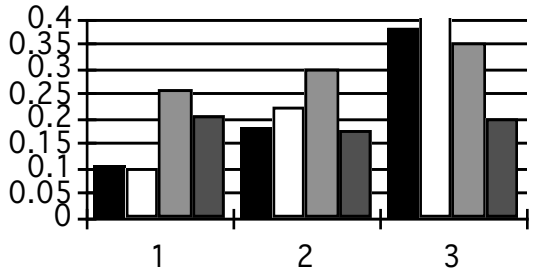

Figure 5.b Position Error at each corner for three consecutive laps around test course when using closed loop odometry. Error is measured at corners of circuit.

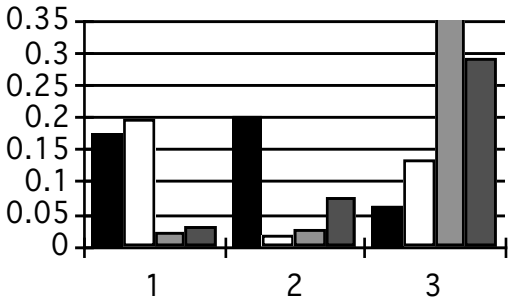

Figure 5.c Position Error (in meters) at each corner for three consecutive laps around test course when using a single camera for navigation (no odometry).

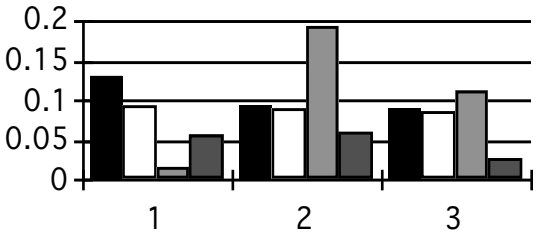

Figure 5.d Position Error (in meters) for three laps when using a pair of parallel cameras for navigation (no odometry).

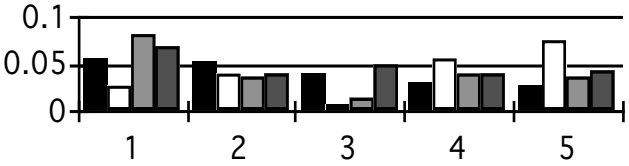

Figure 5.e Position Error (in meters) for five laps with a pair of cameras with divergence at 10° (no odometry).

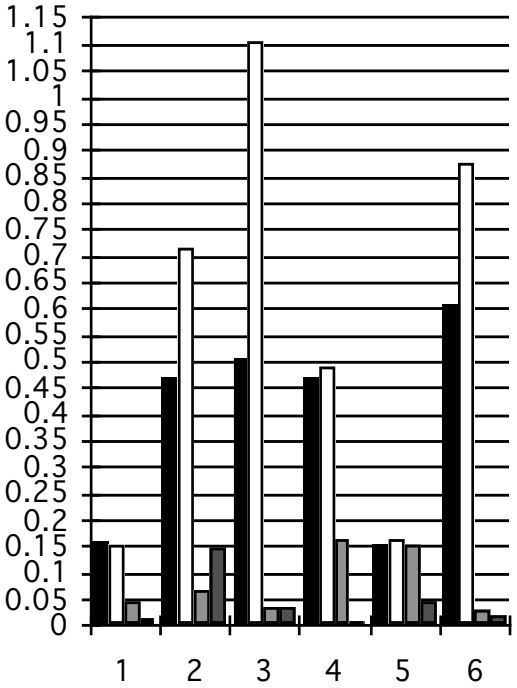

Figure 5.f Position Error (in meters) at each corner for six laps using a pair of cameras with divergence at 20°.

The robot is driven in both open and closed control. The robot controller includes a robust but seemingly imprecise trajectory planner and PID controller. The controller is a black-box in terms of our implementation and consequently we added a meta-controller as a corrective process (interaction following the Nyquist criteria). Using appearance based navigation we contrast the effectiveness of the three different camera configurations.

Analysing the results so far obtained, we can make a number of observations. Both of the odometry based approaches show a continually increasing error. Positional errors accumulate over time and eventually force the operation of the robot to be interrupted. Closed loop control in fact results in a much worse performance than anticipated, this we attribute to a combination of the larger number of small manoeuvres performed (increased slippage), problems properly satisfying the Nyquist criteria for the meta-controller, leading to instability and the fact that having used a closed test trajectory some errors in the base controller compensate each other.

Using vision for navigation, error in position is much better managed; images feeding back into the loop to control and correct positional drift. The paired cameras in particular, demonstrate an encouraging stability with no noticeable increase in error over time. The single camera approach is less stable, results in fact suggest that when servoing, one camera is adequate as most of the error is derived from the difficulty in detecting the stop position before performing turning the corner and not in entering the corner at the correct angle. In experiment, Vision-4, an analysis of the variance in x and y for a robot centred frame showed that for the corners with a long approach variance in y was four times that of variance x, support for the observation that the difficulty is not in servoing but in stopping.

4.1. Single or paired cameras

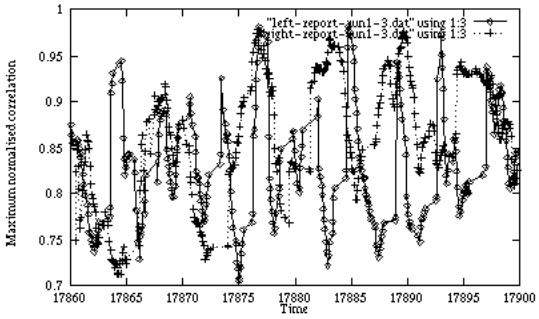

Figure 6. Excerpt from the correlation trace of a pair of cameras, showing the improvement in the maximum correlation.

Using a pair of cameras we introduce redundancy and add considerably to the robustness and reliability of the systems as a whole. Improved precision (see Figure 6) leads to an improvement in performance that brings with it increased reliability. Robustness is evident when we consider that should the correlation process from one camera fail entirely navigation may continue with just one camera until the process can recover. Failure of both correlation processes stops the navigation, however should either sub-process recover from failure, navigation will be resumed.

The improved precision is a result of a better overall performance in terms of correlation. An extract from a runtime trace shows that during execution the maximum correlation switches between left and right images. Over time, the pair of images provide a maximum correlation that neither can achieve individually.

Using a pair of cameras the question arises as to the possibility of improving the accuracy with which goal points are detected by offsetting the cameras and widening the field of view. Offsetting the cameras at 10° from the axis of forward movement, the fields of view overlap at 1m, in a configuration that experimentally gave the best results (Fig. 4e. However, at 20°, there is a marked drop in performance, particularly in terms of stability. We attribute the worsening of performance to a combination of factors. Images change more quickly when the cameras are posed at angle to the direction of travel, this can make correlation difficult and unstable. A high correlation in such a configuration is more precise in terms of position but more difficult to obtain, the target is smaller and better defined. A second factor to consider, is that the scenes presented to the two cameras can be very different, one from the other, fields of view no longer overlap and typical image characteristics differ. Angled cameras also challenge the calibration of the turning angle, as the image plane is no longer typically parallel to walls etc. in the environment making it difficult to define a relationship between the offset of the correlation peak and a corrective turning angle. It may be possible to add a PID loop, including feedback terms in order to modify gains and so provide online adaptation.

4.2. Synchronised and asynchronised databases

When using a pair of cameras there is the question of whether the databases for each camera should be synchronised or not. In other words, should we expect the correlation of images to be independent of the view of the environment. The correlation curve of maximal correspondence over time for a trajectory demonstrates a frequency that is dependent on the information and structure of the environment in the field of view. A highly structured environment with the camera at an angle leads to more rapidly changing images and the requirement for a larger database. A high density of images for a trajectory can leads to instability in the choice of current image. Instability in the choice of current image makes it difficult to identify goal positions.

If we consider a scenario where the robot navigates along the side of a room, with one camera directed into the room and one towards the near wall. The maximum correlation curve over time differs considerably between cameras, as one camera views the rapidly changing features of the near wall and the furniture alongside, and the other looks into the room and views more distant stable features. A zoom could compensate a certain amount of this disparity, but a more general solution is to provide asynchronous databases of images, so that database images are taken for each camera as they are needed rather than at a compromise position or when required by one camera. If we force a lower frequency process to reinitialise so as to be in synchronisation with a higher frequency process, there is a flattening of the maximum correlation curve and density of data on the trajectory that introduces instability in the choice of current image. Instability in the choice of current image makes it difficult to identify goal points.

Parallel or sightly diverging cameras have a very large overlap in their fields of view and so may effectively operate with a synchronised database. In an early test we used the database of single centrally located camera as the

data for two parallel cameras, which together successfully servoed on the target. However, the problem of synchronised images will recur whenever the view of the environment differs considerably from one camera to the other.

5. Conclusions

An appearance based approach to navigation was successfully applied for a number of different camera configurations. In the optimum configuration for our test task a paired camera approach performed excellently, repeatedly navigating to within centimetres of a target point. The approach as a whole performed much better than simple odometry based strategies, using images the robot is continually relocalising itself within the environment and so able to correct positional drifts and compensate for the failings of internal models.

In optimum configuration the paired cameras demonstrate an excellent robustness through redundancy. ZNCC allows for a certain variation in lighting and , occlusion. Pairing the cameras improves the overall performance of the system, avoiding some problems, and allowing for the complete failure of a camera or correlation process.

The visual process architecture has demonstrated some of its flexibility: all of the test applications were built from a common set of elements. Processing has been effectively distributed over the local network, supporting the local hardware configuration and improving performance. For tracking/servoing with correlation we achieved a 12Hz frequency of processing which translates into a movement of the robot of approximately 0.05m between images. At 12Hz we can effectively follow the trajectory applying steering angle corrections as required, sure that the robot will only ever move between consecutive images in the database. A lower frequency of image processing would permit the robot to drift further from its course before being corrected and complicate processing with the possibility that database images may have been skipped. The impact of a synchronous vs asynchronous and centralised vs distributed processing is discussed in the doctoral thesis [Jones 97].

The experiments have also served to demonstrate the different nature of corridor and room environments. The variation of obstacles, surfaces and textures is much greater within a room than within a corridor. It is this variation that limits the usefulness of sonar and which also challenges the use of appearance based techniques. Correlation relies on there being just sufficient detail in the images so that the correlation peak is clear and well defined. Environments with too little or too much structure may in equal measures pose the problem of an indistinct correlation peak, making servoing difficult. The issue of a synchronous/asynchronous database is less of a problem in a corridor than in a room where open space may greatly contrast with a nearby wall. Future work would combine sonar, vision and odometry in robust reinforcement of the navigation task.

Bibliography

[Crowley-Christensen 94] J. L. Crowley and H. I Christensen, Vision as Process, Springer Verlag, Heidelberg, 1994.

[Crowley-Bedrune 94] J. L. Crowley and J. M. Bedrune, "Integration and Control of Reactive Visual Processes", 1994 European Conference on Computer Vision, Stockholm, may 94.

[Crowley-Martin 95] J. L. Crowley and J. Martin, "Experimental Comparison of Correlation Techniques", IAS-4, International Conference on Intelligent Autonomous Systems, Karlsruhe, March 95.

[Inoue et al. 92] H. Inoue, T. Tashikawa and M. I. Inaba, "Robot vision system with a correlation chip for real time tracking, optical flow, and depth map generation", The 1992 IEEE Conference on Robotics and Automation, Nice, April 1992.

[Jones 97] S. D. Jones, "Robust Task Achievement", Doctoral thesis, I.N.P. Grenoble, March 1997

[Matsumoto96] Y. Matsumoto, I. Masayuki, and H. Inoue. "Visual navigation using view-sequenced route representation.", In Proc. of the IEEE Int. Conf. on Robotics and Automation, pages 83-88, Minneapolis, Minnesota, Apr. 1996.

[Schiele 96] B. Schiele and J. L. Crowley. "Object recognition using multidimensional receptive field histograms.", In European Conf. on Computer Vision, pages 1039-1046, Cambridge, UK, Apr. 1996.

[Pentland-Turk 91] Turk, M. and Pentland A., “Eigenfaces for Recognition” Journal of Cognitive Neuroscience, 3(1):71-86, 1991.

[Zoppis 97] B. Zoppis, "Outils logiciels pour le contrôle et l'intègration en vision par ordinateur", Thèse Doctorale, I. N. P. Grenoble, 1997.