

Feasibility of active vision for inspection of continuous concrete pipes.

BRAMMER, Karl.

Available from Sheffield Hallam University Research Archive (SHURA) at:

<http://shura.shu.ac.uk/19393/>

This document is the author deposited version. You are advised to consult the publisher's version if you wish to cite from it.

Published version

BRAMMER, Karl. (2003). Feasibility of active vision for inspection of continuous concrete pipes. Masters, Sheffield Hallam University (United Kingdom)..

Copyright and re-use policy

See <http://shura.shu.ac.uk/information.html>

CITY CAMPUS, HOWARD STREET I
• . " IEIDS11WB

101 715 972 6

CITY

Fines are charged at 50p per hour

0 7 NOV 700b

REFERENCE

ProQuest Number: 10694274

All rights reserved

INFORMATION TO ALL USERS

The quality of this reproduction is dependent upon the quality of the copy submitted.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if material had to be removed, a note will indicate the deletion.

uest

ProQuest 10694274

Published by ProQuest LLC(2017). Copyright of the Dissertation is held by the Author.

All rights reserved.

This work is protected against unauthorized copying under Title 17, United States Code
Microform Edition © ProQuest LLC.

ProQuest LLC.
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106- 1346

Feasibility of Active Vision for Inspection of Continuous Concrete Pipes

Karl Brammer

A thesis submitted in partial fulfilment of the requirements of
Sheffield Hallam University
for the degree of Master of Philosophy

April 2003



Abstract

This thesis describes work to establish the feasibility of using active vision on a mobile robot to improve survey techniques for concrete and clay sewers of less than 1m diameter. Software and hardware components of a prototype mobile remote visual sensing system have been designed and developed.

The active vision system (AVS) operates within smooth-walled small-bore pipes ($0.5\text{m} < d < 1.0\text{m}$). The AVS consists of two distinct, but related hardware components, a controllable (pan and tilt) camera head mounted on a remote control tractor and a system control unit which interfaces this remote system to a PC-based system supporting image capture and analysis.

The software associated with the AVS comprises modules to control the camera orientation and supplement existing Artificial Intelligence vision analysis tools. The latter modules estimate the vanishing point (VP) of a sewer pipe (as a reference feature) and detect coaxial cracks in the periphery of the image (nearest the camera). Control software for the camera head has also been developed.

The VP detection and crack detection modules have been evaluated on images captured from library videos of sewer surveys. The results show that the routines successfully locate the VP and can successfully detect coaxial cracks in a predefined region of interest in an image. The AVS as a whole has been tested in a laboratory setting using a short section of concrete pipe and simulated cracks in its wall. The AVS successfully implements a control cycle which determines and fixes the pipe VP, detects coaxial cracks in the pipe wall, orients the camera to attend to those cracks, and then re-fixes the VP.

Table of Contents

Page		
1	1	The Sewer Maintenance Problem - An Introduction
1	1.1	Sewer Construction and Condition
3	1.2	Common Failure Modes in Concrete Sewer Pipes
5	1.3	Current Survey Techniques for Sewers
5	1.3.1	Camera, Video and Illumination Systems
6	1.3.2	Cables and Traction
6	1.3.3	Display and Recording Units
6	1.3.4	Human Factors
6	1.3.5	Survey Productivity
7	1.4	Recent advances in survey technology
7	1.4.1	Infrared Thermography
8	1.4.2	Sonic Distance Measurement
8	1.4.3	Ground Penetrating Radar (GPR)
8	1.4.4	Integrated sensor systems
9	1.5	Summary
11	2	The Research Project
11	2.1	Long-term Project Vision
11	2.1	Project Scope
13	2.3	Project Aims and Objectives: Active Vision for an Autonomous Inspection Vehicle
15	3	Literature Review
15	3.1	Active Vision
20	3.2	Feature Detection
20	3.2.1	Image Processing
22	3.2.2	Pipe Joints
23	3.2.3	Pipe Junctions
24	3.2.4	Pipe Flaws, Cracks and Fractures
25	3.2.5	Vanishing Points
29	3.3	Recently Published Research
30	3.4	Summary
32	4	Realising the Prototype Active Vision System (AVS)
32	4.1	Requirements for Prototype AVS
33	4.2	System-level Control Architecture
34	4.3	Active Vision System (AVS) Hardware Architecture
36	4.3.1	The Pan/Tilt Head.
37	4.3.2	The CCD Camera
37	4.3.3	The AVS Control Unit.
39	4.3.3.1	The 2-Axis Stepper Motor Programmable Controller Board
40	4.3.3.2	Stepper Motor Drive Cards (Cards 3 and 4)
40	4.3.3.3	HITACHI Alpha-Numeric Display (Card 5)
40	4.3.3.4	Opto Coupler Input and Output (Card 6)
41	4.3.3.5	Switch Mode Power Supply Unit (Card 8)
41	4.3.4	Tractor Unit
42	4.4	AVS Software Architecture
42	4.4.1	Control Module
42	4.4.1.1	Mechanical Constraints
43	4.4.1.2	The Control Element of the AVS.
44	4.5	Image Analysis Software Modules
44	4.6	Hardware and Software System Summary

46	5	Detecting Vanishing Points in Images of Concrete Sewers
47	5.1	Intensity-based Vanishing Point Detection
49	5.2	Vanishing Point Estimation from Isoluminance Contours
49	5.3	Evaluation
55	5.4	Discussion of Results
56	6	Crack Detection
57	6.1	Edge Detection
67	6.2	Coaxial Cracks and Crack Centres
72	6.3	Summary
73	7	Laboratory-based Prototype Testing
73	7.1	Test Rig
75	7.2	Image Analysis Tests
75	7.2.1	Initial Qualitative Evaluation
77	7.2.2	Quantitative Evaluation for VP Estimate
78	7.3	AVS Control
79	7.3.1	Recovery of VP alignment
82	7.4	Orient Camera Axis to Crack Centre
84	7.5	Summary
85	8	Conclusion
89		References
A-1		Appendix 1 Evaluation of Bespoke versus Commercial AVS System
A-2		Appendix 2 Camera Head Kinematics and LINKLAB Simulation
A-7		Appendix 3 TINA Code

1 The Sewer Maintenance Problem

This chapter provides an overview of the sewer maintenance problem and describes common problems that occur with existing survey techniques. It includes a review of recent advances in survey technology.

1.1 The Sewer Construction and Condition

Sewer systems are a vital, if unglamorous, part of the civil infrastructure supporting modern living. In the late 19th and early 20th centuries there was a rapid growth of the sewer system across the world, especially in urban areas. This trend continues.

The sewer system is broadly characterised by two construction technologies for the transfer of waste water. Older sewers are constructed of brick. More recently, clay or concrete pipes have been used. Clay and concrete sewers are constructed by laying a contiguous series of generally straight pipe elements, usually of circular cross-section, connected by occasional curved elements at corners (Figure 1.1).

Today, the combined effects of ageing, unanticipated demands on capacity, and undermaintenance pose a threat to the integrity of these systems [Wirahadikusumah et al., 1998; NSF, 1993; NWC, 1977]. Failure of sewer systems is inconvenient, often expensive to repair, and can pose a threat to public health. Despite this, it is frequently the case that that maintenance is carried out on a reactive basis when collapse or flood occurs. The impact of the legacy of undermaintenance is not just the cost resulting from catastrophic failure; it is also reflected in decreased performance leading to higher running costs [Wirahadikusumah et al., 1998].

An obstacle to improving maintenance is the fact that our knowledge of the condition of sewers in the UK (and elsewhere) is very limited. Approximately 96% of sewers are of less than 1m diameter and are classified as non-man entry

(NME). In the UK, NME sewers account for over 5000 km of those pipes thought to require inspection or attention. The exact location of many NME sewers is unknown due to incomplete and/or obsolescent records.

In 1977, the UK's National Water Council (NWC), in its report entitled 'Sewers and Water Mains - A National Assessment', suggested that around 10% of the UK's sewers are likely to require reinstatement through the next decade [NWC, 1977]. The NWC urged a rolling programme of sewer renewal and maintenance at an estimated annual cost of £148m. (A similar picture has been found in the USA [NSF, 1993].)

It seems evident that even small percentage savings, brought about through improvements in renovation and maintenance practice, would lead to appreciable reductions in both capital and revenue expenditure. A precondition for such a programme is an effective, efficient and economic survey technology. The work described in this thesis represents a contribution to this effort.

Figure 1.1. Three contiguous sections of a modern concrete NME sewer pipe. Note the flanges where the sections are joined, and the two cast junctions joining the pipe from the left.

The work described here concentrates on clay and concrete sewer pipes as these constitute about 90% of the UK's sewers, by length. Complementary research by my co-workers in the Sheffield Hallam University AI Lab was exploring similar issues related to brick sewers. Brick sewers feature mortar joints between brick courses. In many respects, these mortar joints themselves resemble cracks and fissures. Addressing the problem of crack detection in concrete and clay pipes, where the mortar joints are absent, was deemed a sensible first step prior to tackling the more difficult problem of crack detection in brick sewers. Finally, the choice was also determined by pragmatic considerations as the laboratory facilities available did not allow the construction of a brick sewer, but a section of concrete sewer pipe was readily installed (Figure 1.1.).

1.2 Common Failure Modes in Concrete Sewer Pipes

Cracks and fractures may occur in any position and orientation and have a wide variety of general shapes. However, in both brick and concrete pipes, the majority of structural failures fall into one of a number of categories or modes [WRC, 1986]. These include profile deformation, sag and cracking (followed by collapse), pipe deflection and joint displacement.

In clay and concrete pipes* the common failure modes follow from standardised practice in the laying of pipes. Most pipes are almost horizontal and ground pressure is generally normal to the pipe wall. External pressure can cause pipe joints to be displaced. It also causes individual pipe elements to become deformed, eventually introducing small cracks or larger fractures in the pipe surface. Many such faults are co-axial, i.e. roughly parallel to the pipe's (cylindrical) axis. Figure 1.2(a-c) illustrates some of these common failure modes in concrete sewer pipes.

* Henceforth, the term 'concrete sewer pipe' may be taken to include clay pipes unless stated otherwise.

(a)

(b)

(c)

Figure 1.2. Some common failure modes in concrete sewer pipes: (a) pipe joint displacement; (b) coaxial crack (in roof of pipe) and slight joint displacement; (c) radial fracture near pipe joint and coaxial cracks between the joint and the fracture. (These images were made available by Neil Bunting of North West Water, Chesterfield, UK.)

1.3 Current Survey Techniques for Sewers

Structural surveys are commonly conducted by internal inspection. In larger diameter sewers, direct inspection by a person who enters the sewer is possible. For NME sewers, inspection by closed-circuit television (CCTV) is currently the technique of choice [Porch, 1979]. A camera and lighting unit are mounted on a sled or tractor and pulled by cable through a length of NME sewer pipe (the cable is just visible in Figure 1.2(c)). The CCTV camera is connected to an aboveground survey station via an umbilical cable that relays the camera control signals from the ground station to the camera and the image signal from the camera back to a display unit and video recorder. The video images are examined by an operator who identifies and records salient features [WRC, 1986]. The quality of current CCTV systems is variable. A number of factors, technical and human, contribute to overall performance. These are considered below.

1.3.1 Camera, Video and Illumination Systems

The two factors of importance in respect of cameras used for pipe inspection are quality of image and robustness. The first factor encompasses parameters such as resolution, signal to noise ratio, contrast and distortion. Many systems use low quality cameras which give poor images. The images are usually recorded and viewed as a (VHS) video stream, a medium which itself irretrievably loses information from the original image and introduces noise. The quality of the image is also dependent upon illumination and this is often primitive: few tractors have the facility to vary and control lighting conditions in the sewers. Illumination is often configured to aid the human interpreter and this does not always suit the requirements of image processing techniques.

The second factor, robustness, relates to the extent to which cameras are able to withstand the uncertain and often hostile environment within the sewers.

1.3.2 Cables and Traction

Operating (control) signals and video feedback are communicated via cables. There may be separate cables for winching the tractor or sledge forward and backward along the pipe. In some cases the winching and data cables may be incorporated into a single armoured sheath of cable. The connecting cables limit the length of sewer to be inspected. Traditional traction methods are being replaced gradually by self-traction devices where they can be used. Self-traction devices are a comparatively new development in the UK and are not yet used extensively. Current self-traction devices consume a large amount of power. The size of the inspection vehicle may also be a problem. Currently self-traction devices are only suitable for sewers above 225 mm in diameter.

1.3.3 Display and Recording Units

To enhance the value of videotape, facilities are usually available for annotating images at the ground station. Annotation frequently includes distance surveyed, location, image number (index) and date and time. Occasionally, annotation may obscure features of interest in recorded images.

1.3.4 Human Factors

The skill and experience of the members of the survey team contribute significantly to the quality of the survey information. The interpretation of the CCTV images requires selective and subjective judgement and these depend upon prior experience and familiarity, mental awareness, field distraction and interruption. The task of capturing and recording CCTV footage is difficult and many teams concentrate on this aspect of the task. Attention is paid to obvious and significant evidence of impending failure. Substantial libraries of video have been accumulated, awaiting closer scrutiny than was possible in the field. However, such 'off line' scrutiny is time consuming, tedious and expensive [Wirahadikusumah et al., 1998a].

1.3.5 Survey Productivity

Current survey techniques are time consuming and, as indicated above, the earliest signs of deterioration may well be missed given the difficulties of operating in the field. For each defect noticed the operator stops and scrutinises the images [Moselhi and Shehab-Eldeen, 1999]. This may involve time in manoeuvring the sled or tractor to try to improve the image. The effort involved in deploying and redeploying the tractor and cables in contiguous sections of pipe is significant.

1.4 Recent advances in survey technology

The research challenge posed is to develop systems against the following performance related criteria [Wirahadikusumah et al., 1998]:

- i) more accurate and dependable survey information
- ii) improved survey efficiency and economy
- iii) reduced disruption from survey
- iv) ease of deployment of surveying technology

Recent research has both extended the range of technologies used and improved on CCTV techniques*. Some projects have sought to integrate information from different sensor systems.

1.4.1 Infrared Thermography

Infrared thermography is a surface technology that detects thermal energy gradients in the ground [Weil et al., 1994]. Diagnostic gradients arise if either the pipe wall or the insulating backfill around it are compromised. It has proved effective at locating leaks and voids caused by erosion. The technology is

* The developments described have all been reported since the inception of the research programme that forms the substance of this thesis.

relatively easy to deploy and so the production rate is high (3-100 miles per day). However, any positive diagnoses require further complementary survey techniques and so this production rate is perhaps misleading. The major disadvantage is that interpretation of the thermal images is highly skilled and so the percentage of false positive and false negative diagnoses may also be high.

1.4.2 Sonic Distance Measurement

Sonic distance measurement deploys a sled or float-mounted tool within the pipe [Price, 1995]. A piezoelectric transducer generates a sound signal, the timed returns of which are then detected. Given that the velocity of sound changes with density and elasticity of the transmitting medium (e.g., air, water, slurry, concrete, aggregate, soil) a radial profile of the sewer void, the pipe wall, its insulation and backfill can be formed. The system is comparable to conventional CCTV technology in respect of ease of deployment and productivity. Its strength is perhaps in detection of early symptoms of failure such pipe wall deflection and corrosion.

1.4.3 Ground Penetrating Radar (GPR)

GPR deploys an antenna, mounted on a robot tractor, within the sewer [Kuntze et al., 1995]. Electromagnetic waves are transmitted and the reflected waves contain information about the electrical properties of the interfaces between transmitting media. Signal analysis reveals defects in sewer structure and the conditions in the matrix surrounding the sewer pipe. Again, the system is comparable to conventional CCTV technology in respect of ease of deployment and productivity. Its strength is in the information about the condition of the surrounding soil, e.g., voids and other pipelines/cable systems. Interpretation of GPR data is highly skilled and subjective.

1.4.4 Integrated sensor systems

There are three independent attempts to deploy integrated sensor systems on a single survey vehicle.

KARO [Kuntze et al., 1995] is a prototype robotic inspection system that fuses data from ultrasonic, microwave, 3D optical sensors and colour TV camera technology. It has optional GPR and cartography units. The system is winched through a pipe section. Control signals and data transmissions are communicated via an umbilical cable to a ground station. The main advantage of the prototype is the integration of complementary sensor systems.

PIRAT [Campbell et al., 1995] integrates CCTV with either sonar (flooded pipes) or laser (drained pipes). It is connected to a ground station where the control system and the system for interpreting the complementary signals are located.

The Sewer Scanner and Evaluation Technology system (SSET) [Abraham et al., 1997] is a prototype system combining a scanner, CCTV and gyroscope technology to provide an analysis of faults over a pipe section. In the prototype, the sensor systems are carried on a rigid sled that is winched through the pipe. The designers of SSET claim that its major benefit is that its data logging and analysis facilities relieve the survey technician from the task of analysing the CCTV signals during the survey itself.

1.5 Summary

There continues to be a need for improved sewer survey techniques. The sensor technologies described above are largely complementary. Ease of deployment and productivity (e.g., IR Thermography) are traded against resolution. All survey technologies continue to depend upon teams of skilled and experienced technicians and engineers for their operation and visual inspection will continue to play an important part in fault diagnosis.

The survey units are relatively large and the use of power and signal cables to control them and the use of steel cables to propel them pose appreciable constraints on productivity. Improving productivity will probably entail increasing the autonomy of the survey units. This implies a need for cheaper, more robust tractors and sensors together with signal transmission systems that avoid umbilical cables.

2 The Research Project

This chapter begins by outlining the long-term vision held by a group of research co-workers, based initially at the AI Lab at Sheffield Hallam University and which forms the context within which this work was undertaken. It then defines the scope of the work that concerns this specific project. This leads to a specification of the project aims and objectives of the work subsequently undertaken and documented in this thesis.

2.1 Long-term Project Vision

In Chapter 1, I have suggested that modest, incremental automation of elements in the surveying process offers potential benefits in terms of cost effectiveness, accuracy and safety. New developments in the field of robotics offer the prospect, albeit well into the future, of a step change in the autonomy of survey vehicles. Increasing technological advances in microelectronics and communications offer the prospect of smaller inspection vehicles with cameras and other sensors less constrained by demands for electricity supply. An intelligent autonomous robotic pipe inspection vehicle is not beyond the realms of possibility and its realisation seems a legitimate long-term research goal.

2.2 Project Scope

Surface-based technologies (e.g., IR thermography) will frequently require confirmation by inspection from within the sewer pipe. As we have seen, a variety of sensing technologies may be deployed within the sewer itself. The decision of three independent research projects, KARO, PIRAT and SSET (see Chapter 1), to deploy multisensor systems reflects the fact that such systems can provide complementary data about the sewer pipe and its supporting matrix, to give a fuller picture of sewer condition than internal inspection alone. However, all three projects use vision systems to support inspection. CCTV is cheap and requires relatively little expertise to interpret data. Visual inspection gives engineers the confidence to undertake much more expensive and potentially

inconvenient excavation and repair. For these reasons alone, it seems likely that vision will play a role in most future inspection platforms.

A prime motivation for computer vision research is to develop sensor systems capable of supporting robots performing a variety of tasks in both open and enclosed environments (e.g., offices, factories, tunnels, reactors). Hence, anticipating advances in autonomous robotics, computer vision seems likely to play a role in enabling robots to interpret their environment.

Visual inspection is a major research topic in computer vision and attempts have been made (with mixed results) to detect flaws in a variety of materials [Wallace, 1982]. There is, however, little work published on the identification of flaws such as cracks and fractures in either concrete or clay. There is even less work on the problem of visual inspection in sewers. This may be because assumptions, which are valid in many circumstances, are not valid in relation to sewers. For example, inspection systems in other domains assume that the area to be inspected can be viewed from a single viewpoint and so inspection can be performed on the basis of a single image.

In small-bore pipes, the inspection camera is typically orientated so as to provide a view down the pipe: the camera and pipe axes are almost parallel. Coaxial fault lines are therefore heavily foreshortened - points at either end of the image of a fault arise from widely separated points on the pipe surface. As a result, coaxial faults are not readily described from a single image. The resolution of the image of one end of a fault is likely to be significantly different from that obtained at the other end. A suitably high-resolution description of the full length of the fault can only be obtained by integrating data extracted from a sequence of images captured over time as the camera travels down the pipe.

The pipe surface may be assumed, locally at least, to be a cylinder. If the cracks move away from the optical axis (centre of the image) on a cylindrical surface, distortions in the perceived width of the crack will be introduced by perspective. As the pipes we are concerned with have $< 1\text{m}$ diameter, and some may be as

narrow as 300mm diameter, these distortions could be significant. It is proposed that location and accurate description of coaxial faults in small-bore pipes is best achieved by an active computer vision system capable of automatically altering the orientation of the camera. The camera will detect faults in the pipe surface. Eventually, it must integrate data from a sequence of images thus obtained.

One approach to this problem would be to use statistical pattern recognition. This would involve building models from training sets of sample data. However, cracks in sewer walls vary considerably and it is not clear that a reliable statistical model could be built. At the very least, an extremely large training set would be required. This study seeks to establish the feasibility of taking a comparatively simple, structural approach based upon heuristics which describe the features expected of a crack in a pipe wall.

2.3 Project Aims and Objectives: Active Vision for an Autonomous Inspection Vehicle

The principle and practical aim of this research programme is to establish the feasibility of a prototype tractor-mounted Active Vision System (AVS) capable of detecting cracks and fractures on the internal surfaces of concrete NME sewer pipes.

The principle objectives were to design and then evaluate key elements of the AVS prototype. Within this overall objective, other objectives or milestones of the investigation were determined as follows:

- i. review the literature on active vision for sewer and other pipes (especially concrete pipes);
- ii. identify and evaluate algorithms for initial detection of cracks/fractures in small-bore pipe walls given a single image of the pipe;
- iii. determine the camera kinematics required for feature fixation;
- iv. develop the system hardware;
- v. develop the system software;

- vi. test the prototype system on simulated faults in a laboratory environment;
- vii. draw conclusions relating to the feasibility of further development of the prototype AVS.

While a final pipe inspection system should be capable of operating in real time, the current project sought only to demonstrate the feasibility of the proposed approach.

Experimental evaluation was to be performed using laboratory-based facilities. The final tractor mounted vision system was tested within a concrete NME sewer pipe (Figure 1.1) and evaluated on simulated cracks introduced on inner surfaces of the sewer pipe.

The remaining chapters of this thesis address each of the above objectives in turn.

3 Literature Review

The scope of this review is determined largely by a focus on vision research relevant to active inspection of continuous, smooth walled, cylindrical pipes. Section 3.1 examines the relevant work carried out in the field of active vision. It looks first at low-level object detection and tracking, where the detailed description of the object is not of primary interest. It then looks at higher-level object analysis where, for example, the dimensions and orientation of an object are of primary interest.

I have chosen not to include extensive reference to active vision using stereo. Use of a second camera is constrained by the limited room available in NME pipes and second cameras entail additional expense – not least in terms of computational effort. More significantly, the cracks and fissures of interest in this domain are found on the walls of the pipes and, given knowledge of the pipe geometry relative to the camera, and some basic information on distances (e.g. between successive pipe joints) the relative depths of different features may be derived from images taken with a single camera.

Section 3.2 looks specifically at feature detection in concrete sewer pipes. The features of interest in this domain are pipe joints, pipe junctions and pipe flaws such as cracks or fractures. This section also examines work that seeks to determine the vanishing point (VP) within a pipe. (The VP may be used as a valuable reference point for visual analysis – see section 3.2.5).

Finally, section 3.3 summarises relevant research that has been published since the conclusion of the experimental research documented in this thesis.

3.1 Active Vision

Active Vision Systems receive much attention from the computer vision community. Much of the work in this area is orientated towards the design of object tracking systems in which a camera actively follows an object of interest.

Allen [1989] describes a motion tracking system capable of identifying and tracking an object wandering in a confined space. The system computes an estimate of the location of the centroid of the robot's motion energy and uses this to guide a camera mounted on a robot arm above the test area. The camera is at a fixed height above the object and so the system tracks in 2D. The dark rectangle of the robot's form is the only object in the image and is relatively easy to identify in a well-illuminated environment against a uniformly white background. Allen highlights the fact that many approaches to motion detection contain "a burdensome computational cost that precludes real-time implementation".

Allen's tracking of a derivative feature (in this case, the centroid of motion energy) seems a useful strategy for simplifying the tracking of an object when its precise orientation, or indeed boundary, are not of immediate concern.

Papanikolopoulos et al. [1992] describe a monocular, real-time, 3D tracking system in which a camera mounted on the end-effector of a PUMA robot tracks an object of interest attached to the end-effector of a second PUMA. The system tracks "distinct features" of an object as selected by a human operator: typical objects were books or pencils. Papanikolopoulos does not expand upon what is meant by the term distinct feature.

In the domain of sewer inspection, some features are readily described as distinct, e.g., pipe joints. It may also prove possible to define some features of cracks or fractures as 'distinct' for the purposes of tracking. However, it would not serve the purpose of this project if such features had to be nominated by a human operator.

An interesting aspect of Papanikolopoulos's approach is the strategy of reducing the computational burden by using world-knowledge. For example, the change in depth of the target between two time instances is constrained by knowledge of the maximum permitted translational and rotational velocities of the robot end-effectors.

Murray et al. [1995] are similarly concerned with target acquisition and tracking in real-time. Again, the goal of their system is to keep a camera focussed on a moving object over extended time periods. The system employs a surveillance strategy that implements a high-level gaze controller as a finite state machine (FSM). The FSM has five states labelled 'inactive', 'wait', 'saccade', 'pursuit' and 'panic'. The gaze controller determines which visual state-related process to use depending upon the visual observations and the current state. The 'wait' state is associated with a general, wide-area search function. If an object of interest enters the periphery of the scene the gaze-controller enters the saccade mode, causing the camera to centre (foveate) the target object. If analysis confirms identification of a target, the 'pursuit' state is entered. Loss of the target (usually because its movement violates an assumed constant velocity) triggers the 'panic' state, which attempts to predict a new location for the target. If the target is not quickly recovered, the 'wait' state is re-engaged.

Their implementation of the vision analysis states, in conjunction with the mechanical camera control feedback, achieves responses to a peripheral target or a tracked target quickly enough to satisfy the 'real-time' requirements of their problem domain, which is tracking human movement. It is not specified how the system deals with multiple features of interest entering the periphery.

The fault detection problem in sewers differs from the tracking and fixation problem in that, instead of attempting to solve low-level tasks in real time, we must address a higher level problem - active visual inspection using necessarily simpler hardware. However, the state-based approach to active vision adopted by Murray et al. does capture, in a natural way, the general problem of control in a closed loop vision sensing system. Each active state is associated with a system function. Camera control signals, as actions, are associated with state transitions. The active FSM approach of Murray et al. would seem to provide the basic structure for an active vision system for sewer inspection.

Brunnstrom et al. [1996] have attempted higher-level active visual inspection. They have confined themselves to what they call a “static world containing man-made objects”. Following Malik [1987], they define objects as “opaque solid objects bounded by piecewise smooth surfaces with no markings or texture”. Their goal is the identification of manufactured piece-parts. They assert that “in an active, continuously operating vision system it is not reasonable to base recognition on complete surface or scene reconstruction”. Instead, they propose using ‘selective attention’, ‘fixation’ and ‘structure integration’. Attention and fixation are distinguished: attention relates to a region of the image, fixation relates to the zoom and focus operations that identify points within a feature, e.g., a point on a curved line. Their approach features a number steps; find a structure to attend to, fixate the structure, choose another attention point and decide whether re-fixation is required, decide whether junctions should be connected, integrate structure information between one fixation and another. Attentional features of interest are ‘standard’ shapes such as junctions of straight and curved edges. The camera fixations are determined by a grouping strategy, which forms sets of junctions, separated by depth of field differences (determined with the aid of stereo images). The work has begun the development of a system for rapid active detection and classification of these junctions by selecting fixation points.

The difference between this research and sewer pipe inspection is that in sewers we must deal with both standard and non-standard features. Whilst non-standard features (e.g., cracks) may be classified at some level of abstraction, they are essentially unique. This is mainly because they are naturally occurring and are not man-made. Further, Brunnstrom et al. are looking at object recognition within open and illuminated environments whereas sewer pipes, by contrast, are characterised as relatively small, enclosed spaces and are, by their nature, not well illuminated.

A number of similarities between the two research problems are identifiable. It is certainly possible to share Brunnstrom’s scepticism that ‘recognition based on complete surface or scene reconstruction’ will prove feasible in the case of sewer images – not least because of the signal:noise ratio in the sewer images. As with

Murray et al., the state-associated processing of images seems natural. The concepts of attention and fixation are usefully distinguished. In the case of sewers, joints and cracks may be detected initially by a region-based attentional mechanism whilst the detailed description of these features warrants, in part at least, the closer scrutiny associated with 'fixation'. The idea of defining a set of classifiers would seem to offer a way forward in developing a sewer inspection system, but this too may prove difficult as it is far from obvious that there is a 'standard' set of features such as is found in the junctions of regular polyhedra (pipe junctions may prove an exception). Feature integration may prove more demanding when the objects of interest do not have straight or smoothly curved features.

Some workers report on active stereo vision. Most stereo vision systems use knowledge of the geometry of a pair of cameras to recover depth information from the two slightly separated views they receive from a single scene.

Pretlove and Parker [1993] have developed an Attentive Robot Vision System intended for use in a traditional manufacturing environment. Their system takes advantage of good uniform ambient lighting, which is not available in a sewer environment. In common with many stereo systems, significant computational resources (transputer arrays) are required.

Abbott and Ahuja [1992] have also developed a stereo active vision system which employs two high-resolution cameras for image acquisition. The system is capable of automatically directing movements of the cameras so that camera positioning and image acquisition are tightly coupled with visual processing. The system was developed as a research tool and is largely based on off-the-shelf components. It is not designed for real-time performance and is currently a research tool.

In principle, a single moving camera can be used to generate a pair of images for stereo analysis. In practice, this requires precision control of the positioning of the single camera. This is not feasible in a sewer pipe environment where, for

example, wheel slippage can introduce significant errors in estimates of change in position of the camera/tractor unit.

Stereo vision adds complexity and computational overheads, which we choose to avoid. Further, knowledge of the sewer pipe geometry allows recovery of much relative depth information on the basis of a single image. This is especially true of features such as coaxial cracks, which are of immediate relevance in this project.

3.2 Feature Detection

Interpretation of digital images captured during sewer surveys has recently received considerable attention. The information contained within these images is needed for automated, offline, surveying or active surveying and/or guiding robot vehicles engaged in renovation and repair work within NME sewer pipes. Image interpretation is about relating identifiable features of the image to interesting features of the viewed world.

This section considers different features of interest in turn, prefaced by a short section on edge detection. In addressing a considerable body of literature on feature detection, I have scoped the task by defining an 'environment specific' feature set and seeking related work. Features of immediate relevance to this exploratory work in sewer pipes are normal and abnormal pipe joints, pipe junctions, and other pipe flaws (of varying degrees of severity). In addition to these diagnostic features, the vanishing point (VP) of a sewer pipe constitutes a valuable reference feature for camera orientation and scene analysis and, therefore, research on VP location is also included.

3.2.1 Low-level Features (Edge Detection)

A vital step in the design of a machine vision system is the determination of relations between the interesting features of the viewed world and measurable

properties of the digitised image. If some feature of a viewed object is to be detected by a vision system, it is clearly important to determine how that feature will appear in the pixel array. Figure 3.1 shows a typical grey level digital image, obtained from a field CCTV survey, of a non-man-entry sewer. Pipe joints appear as bright rings. Longitudinal cracks are usually represented by connected series of near-random line segments. Wall encrustation gives rise to a dappling effect. These real-world changes in luminance give rise to sharp changes in image intensity. Step changes in image intensity, commonly known as edges, are easily detected features that may be used as cues to the location of features of interest.

Figure 3.1 Grey-level digital image, obtained from a field CCTV survey, of a non-man-entry sewer. Three pipe junctions and a co-axial crack (top right quadrant) can be seen.

There are a number of edge detection algorithms. Figure 3.2 shows the result of applying one such algorithm, Canny edge detection [Canny, 1986], to the video image of Figure 3.1. (A fuller description of Canny edge detection as used in this project is given in Chapter 6.)

Having generated such an image, the problem remains of distinguishing one feature from another and assessing the diagnostic implications of each.

```
l r j                               mono
  Size   Mouse   Proj   Paint
install  clone   init   repaint
roi: adjust / toggle / show
```

Figure 3.2 Canny Edge Detection on the image of an NME sewer Pipe. Note the edges (in colour) formed by each of the first three pipe joints and by the crack in the top right quadrant. Note the spurious short 'strings' of edge data generated by illumination of wall encrustation.

3.2.2 Pipe Joints

Pipe joints within prefabricated concrete sewer pipes normally occur at fixed distances along the length of the pipes. The pipes are constructed so that all joints are tight fitting and flush. With the passage of time, earth settlement, disturbance

and various other natural or induced effects, pipe joints are displaced to varying degrees (Figure 1.2(a)).

Ruiz-del-Solar and Köppen [1996] propose a neural architecture for an image processing system that recognises circular features such as pipe joints. The system is not evaluated and does not assess the condition of the joints.

Pan et al. [1994; 1995] have developed an approach to the detection of pipe joints. Detection is facilitated by the fact that there is often strong reflection from pipe joints and they are approximately circular. However, reflection from a pipe joint can vary considerably and much of it may be obscured by water and debris in the pipe. Following application of a standard edge detection algorithm, edges that appear to be from a common arc are grouped together. Circle fitting algorithms are then applied to the arc segments. For a healthy joint, the fitted circles should share a common radius and common centre. When the joint has been displaced the radii are common but the centres are displaced.

3.2.3 Pipe Junctions

Pipe junctions occur within sewer pipe systems, when other pipes merge with existing pipe systems. The junction is usually cast as a feature of a special pipe length.

Lateral intersections (see Figures 1.1 and 1.2(b)) generate light image regions, usually smaller and less bright than those arising from similarly distant displaced joints. The near edge of a lateral connection is marked by a sharp change in depth or distance from the viewer as the sewer wall gives way to the incoming lateral wall. The far edge exhibits a similar though reversed change in surface orientation.

Taylor et al. [1988] use Canny edge detection to extract descriptions of lateral pipe connections. The effects of different types of illumination are looked at, e.g., direct or reflected illumination.

3.2.4 Pipe Flaws, Cracks and Fractures

Pipe flaws also include major features such pipe collapse, pipe obstruction, tree root intrusion. These catastrophic flaws are usually detected by virtue of the fact that they bring a survey to a halt!

Less obvious features that may represent the early symptoms of later problems include corrosion of concrete and pipe wall deformation. Pipe deformations, prior to cracking, are difficult to detect from CCTV inspection alone. Pipe deflections of as little as 1% or 2% will result in cracking of clay pipes. Such deflections are readily masked by image distortions of up to 10% introduced CCTV technology.

Henry and Luxmoore [1996] have developed a visual profiling system for CCTV pipe inspection cameras. The positioning of a light source in front of the CCTV camera allows a 'light ring' to be projected on the internal surfaces of sewer/water pipes. Distortions and degradations within the inner surface of the pipe are identified using pattern-matching software.

Xu et al. [1998] have developed an approach to detecting distortion in pipe walls near pipe joints. A digitised image of an illuminated pipe is obtained. It is "cleaned up" prior to analysis. The characteristic circular section of the pipe joint is estimated by least squares fitting of a circle (LS circle) to the data points associated with a well illuminated pipe joint (see Figure 1). Distortion (prior to cracking) deforms the circle into an ellipse. The aspect ratio of the ellipse, which is proportional to the ratio of maximum and minimum diameters, is compared to that of the reference circle to give a measure of the distortion present. Given the profile of a pipe at a pipe joint, this work significantly improves upon human visual inspection for early distortion (<2%). However, the feature extraction currently identifies only 85% of pipe joints and so a combination of human and machine are necessary if more than 85% of pipe joints must be analysed.

Moselhi and Shehab-Eldeen [1999] have described an approach to processing images from clay and concrete sewer pipes. Following standard edge detection operations, they can provide information on axis lengths (major and minor), angular orientation and elongation of significant features, and cracks in particular. Their longer-term objective is to use this derived data to train a neural classifier to distinguish between displaced joints, cracks, and other flaws.

Xu et al. [1998] have extended the approach which detects profile distortion (described above) to detect cracks near pipe joints. When a coaxial crack appears in a pipe wall, all further deformation occurs as rotations about the crack (the crack is rather like a hinge). A plot of the radius of the pipe (boundary distortion) relative to the reference LS circle shows a sharp change in curvature if a crack is present. The technique can be used only if the original size and position of the pipe are known, or can be estimated accurately. If estimation is used (e.g., LS circle), the technique is very sensitive to the estimated centroid and radius and this is a limitation of the technique.

3.2.5 Vanishing Points

A vanishing point is defined as the image projection of an (infinitely distant) point of intersection of a set of parallel lines. Vanishing points have been exploited in a variety of situations. Fischler et al. [1982] use the vanishing point generated by vertical edges of buildings in urban scenes to ease the identification of other vertical objects. Barnard [1982] exploits vanishing points in the interpretation of perspective drawings of plane-faced objects. A further body of work concerns the use of vanishing points in camera calibration. Beardsley and Murray [1992], for example, present an algorithm to recover camera parameters from vanishing points extracted from images of precisely measured calibration targets. Straforini et al. [1993] use vanishing points to recover the heading of a mobile robot travelling through corridors and offices.

Tai et al. [1992] give a review of vanishing point detection algorithms. Most (e.g., Barnard [1982]) rely on some form of Hough Transform. Lutten et al.

[1994] have argued that Hough transform approaches to vanishing point detection are inherently biased towards solutions which lie within the boundaries of the input image. They go on to propose a probabilistic method which avoids this bias.

In sewer pipes, the vanishing point can be thought of as an infinitely distant point on the principal (cylindrical) axis of the pipe. Photographically, it corresponds to the centre of the 'dark' region at the end of the pipe (see Figure 3.1). Although of no diagnostic significance itself, the VP provides a valuable reference point or datum for image analysis and camera orientation. Having identified the VP, it is easier to identify the main structural features of the environment.

Taylor et al. [1998] have proposed a method for estimating the VP within the image of a concrete/clay sewer pipe. This method uses thresholding. A single luminance threshold segments the image into dark and light regions. The largest connected dark region is found and its centroid is taken as an estimate of the VP. The approximation to the actual VP obtained is used to give an indication of where in the image a lateral pipe joint may be found. Pridmore et al. [2000] extend this approach to estimate the orientation of lateral pipes. The advantage of the method of Taylor et al. is that it is much more computationally efficient.

Cooper et al. [1998] have developed an approach to VP estimation in brick sewers. They use a two-stage process. The first stage uses the method of Taylor et al. (above) to obtain an initial point estimate of the VP (i.e., the centroid of the largest connected 'dark' region). The second stage uses the longitudinal (coaxial) mortar lines associated with brick courses (Figure 3.3). A least squares estimate of the intersection of all such horizontal lines is computed. Lines are deemed horizontal if they pass through the neighbourhood of the initial VP estimate. Here, the neighbourhood is defined by a circle, centred on the estimated VP, and with a radius determined as the distance of the furthest pixel in the connected region from the estimated VP. Cooper et al. have gone on to extract information on camera orientation relative to pipe axis when evaluating camera motion during

surveys. The reliance of this technique on visible co-axial mortar lines means it is not applicable for concrete sewers.

(C)

(d)

Figure 3.3. Two-stage process for VP estimation in brick sewers, (a) thresholded image of brick sewer; (b) initial estimate of the VP as the centroid of the largest connected region in the thresholded image; (c) selection of horizontal mortar lines passing through the neighbourhood of the initial estimate of the VP; (d) revised (least squares) estimate of VP.

3.3 Recently Published Research

L. Paletta et al. (1999) have developed a method for automatically detecting inlets in a sewer from video images in real time [Paletta & al., 1999]. Exploiting prior knowledge of sewer pipe construction, they extract regions of interest (defined as trapezoids) to the right and to the left of the distant pipe head. A trained neural network identifies any inlet feature within the region of interest. If an inlet is detected, its centre is estimated, together with a confidence value that the feature is, indeed, an inlet. The system has been evaluated in a laboratory environment and achieves a detection rate of 94.6 percent; no false-positives were detected and only two false-negatives.

Kolesnik and Baratoff (2000a) present a method for computing the 3-D orientations of circular structures appearing in sewer images and their distances relative to the robot. The circular features of interest are the boundary of the dark region which surrounds the VP of the sewer and the elliptical profile of lateral pipe junctions. In both cases a brightness threshold is chosen to segment the image. The dark central region around the vanishing point is assumed to be in the central sub-threshold area of interest is selected and inlet pipe features are assumed to be within the above-threshold segment of the image. The resulting two regions of interests are smoothed using median filtering. Edge detection is applied to the filtered image to detect the edgels of any ellipses. A conic section is then fitted to the edgels. The results are used to estimate camera (robot) orientation and to estimate distances within the pipe environment, e.g., distance to pipe inlet. Such distance information is a key element of any autonomously conducted survey.

Kolesnik and Baratoff (2000b) present an algorithm for recovering the distance between pipe joints. The technique uses standard image processing techniques to detect the distinctive circular profile of pipe joints. Knowledge of camera a sewer pipe parameters allows estimates of distances to be made. A distinctive feature of the system is that the robot orients its camera using a laser crosshair projector to ensure close alignment of the camera and pipe axes. As a result, the circular

structures in the sewer project onto circles (rather than ellipses) in the image, thereby simplifying the image processing operations necessary for detecting, locating, and interpreting them (Kolesnik, 2000).

Moving away from feature detection and analysis, this work has been extended to develop a guidance system for a robot [Kolesnik, 2002; Kolesnik and Streich, 2002]. The simplicity of pipe geometry, together with knowledge of the way in which laser crosshairs will appear when projected onto the pipe wall allows very efficient analysis of an image sequence so as to recover the robot's instantaneous orientation and hence guide its navigation.

Most recently, the image analysis work of Kolesnic (above) has been brought together with the camera technology of the the Center for Machine Perception, Czech Technical University Prague to form the basis of the EU funded ISAAC Project [<http://cmp.felk.cvut.cz/projects/isaac/>]. This project, funded for the period 2002-3, has yet to produce public output.

3.4 Summary

The literature review has identified a number possible components for an overall, initial strategy for the problem of crack detection that is the central concern of this project. The process-related 'active' state machine [Murray et al., 1995; Brunnstrom et al., 1996] is a natural representation for the discrete control steps in an inspection cycle. The distinction between attention and fixation [Brunnstrom et al.,1996], the latter involving a reorientation of the camera to 'centre' a feature attended to is also natural in such an active vision system. The method of VP estimation proposed by Taylor et al. [1998] is appealing for its simplicity but its accuracy is untested. The concept of projecting a light-ring onto the wall of the pipe to detect pipe joints [Henry and Luxmoore, 1996] suggests the possibility either of using pipe joints themselves to estimate a VP (modest deformation of joints by the 1%-2% prior to cracking would prove problematic) or of using luminance thresholds themselves as the basis of a VP estimate (thus

avoiding the need to have a joint in the current image prior to VP estimation). The concept of selecting coaxial mortar joints between bricks in brick sewers [Cooper et al, 1998] suggests an approach to the selection of coaxial cracks in sewer walls. An outstanding concern is the detection of highly irregular, non-standard, features such as cracks.

4.0 Realising the Prototype Active Vision System (AVS)

This section describes the work undertaken to construct the prototype active vision system for sewer pipe inspection. Following a high-level statement of requirements, the structure of the account reflects the design and implementation of the following system components:

- i. system-level control architecture;
- ii. active vision hardware;
- iii. active vision software for camera control and image analysis.

4.1 Requirements for Prototype AVS

The AVS must operate within NME sewer pipes. It will eventually be mounted on an autonomous inspection robot but a remotely controlled tractor may be used to evaluate the feasibility of active vision system components. This has mounted upon it a fixed-axis light source and a small digital camera attached to a camera mounting that facilitates controllable orientation of the camera.

The camera provides a signal to image capture and analysis subsystems. The aim of image analysis is to identify features of interest. I shall be concerned only with detection of coaxial cracks and fractures on the pipe wall, near the camera.

Image analysis and camera control can exploit the cylindrical geometry of the pipe to save computational effort when examining each image. Features of interest are most reliably detected when they are near the camera. By aligning the camera axis with the pipe axis, features on the wall of the sewer pipe nearest the camera will appear in the periphery of the image (and not in the centre of the image). By manoeuvring the camera so that the VP is centred in the image (at least to a reasonable approximation), the robot can achieve the required alignment of the camera and pipe axes.

In operation, the inspection robot first locates the vanishing point and adjusts the camera so that vanishing point is centred in the image. Power to the tractor wheels is turned on and the robot moves slowly along the pipe processing images from the video stream as it goes. As the tractor moves along the pipe, it must periodically re-evaluate the position of the VP and make any appropriate camera adjustment. A simple 'time-out' mechanism can ensure this happens. When an image is examined and a possible feature of interest is detected, the robot should halt and bring the camera to bear on that feature for detailed analysis (e.g., length, width, axial orientation).

Following a successful feature analysis, the robot must again locate the vanishing point, orient the camera in that direction, and start the motors which take it forward down the pipe, looking for the next feature of interest.

4.2 System-level Control Architecture

Given the above high-level description of the system, it is possible to specify a system-level control architecture. Following the active finite state machine approach suggested by Murray et al. [1995], Figure 4.1 provides a state-based description of the system control. Once the system is ready (initialised on power-up), the 'Determine VP' state locates the VP, the 'Fix VP' state then orients the camera towards the VP and the robot is ready to move forward. The robot then enters the 'Detect Feature' state. Moving forward, it processes the image stream looking for features of interest. Periodically, it must check camera alignment, re-entering the 'Determine VP' state. When a feature is detected, the system enters the 'Fix Feature' state, which determines a new orientation for the camera so that the feature can be analysed in more detail. The 'Analyse Feature' state develops a description of the feature (length, width, axial orientation). When complete, the system reports or logs the analysis before re-determining the VP in readiness for the next feature search cycle.

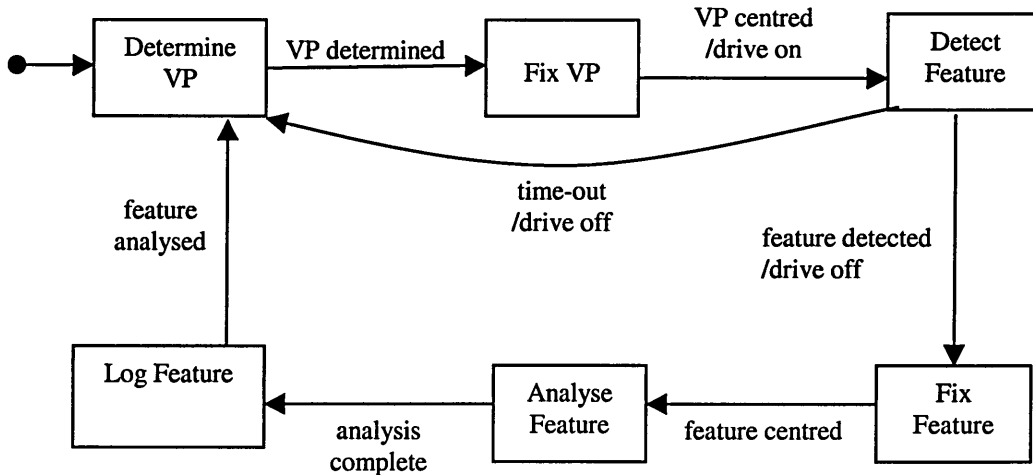


Figure 4.1. Statechart specifying system-level control of active vision system. Each state is an active state, associated with the performance of a system operation.

4.3 Active Vision System (AVS) Hardware Architecture

The AVS requires a controllable pan and tilt head that will support a miniature CCD camera. This assembly must be mounted on mobile robotic inspection vehicle. Control and communications software is required. Constraints on the design of the prototype AVS are specified in Table 4.1.

An evaluation of the relative merits of purchasing a commercial pan/tilt head and controller against design and development of bespoke hardware and software was undertaken. The result was a decision to develop a bespoke system. (Appendix 1 provides further detail of the evaluation.) The details of the bespoke system constructed are presented below.

Table 4.1 Constraints for the prototype AVS hardware.

FEATURE	REQUIREMENT(S)
Weight	Pan/tilt head plus camera must be sufficiently light to be used on a mobile robot system.
Pan/tilt control and flexibility	Hardware must be designed to minimise restriction on orientation.
Positional accuracy	Adequate positional information of the CCD camera (choice of optical shaft encoders, potentiometers and stepper motors).
Back-Lash	A common problem with pan/tilt heads - need to minimise in order to fix new target points efficiently.
Power consumption	The prototype may use a normal power supply. However, there is an eventual requirement for the system to run autonomously from a battery system so power consumption must be kept to a minimum.
Hardware Cost	There was a budget limitation of £700-£800.
Software	Specification, design, programming and testing for the control, interface and input/output capabilities.
Processing	Processing for the control mechanisms would have to be kept to an absolute minimum. There would already be a major processing overhead for analysis of the video stream.
Display	A display giving information on positions of motors and of the commands, etc that had been entered would be advantageous.
Portability	A system that was designed for use by other robotics/AI applications would be extremely desirable.

4.3.1 The Pan/Tilt Head.

The design of the head was developed using AutoCAD (version 12). The casement for the pan/tilt head was Dural, a type of aluminium. This provides a chassis for two stepper motors that control the orientation of the camera.

Four-phase hybrid stepper motors were used in preference to the permanent magnet types as they offered much higher working torque and better stepping rates. These factors were of significance, as the motors would need to move the CCD camera and fittings. One of the motors would need to move the entire weight of the head. The chosen motors were capable of delivering a high detent torque even when not energised. The detent torque of 5mNm, and a holding torque of 70mNm, make this motor capable of moving the Pan and Tilt Head comfortably.

The motors had a step angle of 1.8 degrees. Half step mode gives 0.9 degrees per step.

The weight of each stepper motor is 187g. The remaining weight of the housing for the head was engineered to ensure a total weight of less than 1kg as required.

The chosen stepper motors has a rated voltage of 5V, and a rated current of 0.5A. These allow the motors, and the rest of the electronics, to be powered from a single dedicated power supply based in a controller unit.

4.3.2 The CCD Camera

A commercial camera from WATEC CO Ltd, Kawasaki, Japan was chosen. The WATEC model WAT-902A CCD camera was chosen for weight, cost, resolution and power consumption. It was supplied by Vista Vision Systems (Levanroy) Ltd. The pan/tilt head and camera assembly is shown in Figure 4.2.

Access to Stepper
Motors.

CCD Camera

Pan/Tilt Camera
Mounting

Figure 4.2. Pan/tilt head assembly with WATEC CCD Camera.

4.3.3 The AVS Control Unit.

The Active Vision System Control Unit is housed in a standard, modular, nineteen inch, Euro-card sub-rack system (Figure 4.3).

Figure 4.3 AVS Control Unit.

The Control Unit comprises the following:

- Stepper Motor Programmable Controller - Card 1
- Quadrature Track-ball - Card 2 (development purposes only)
- Stepper Motor Drive - Cards 3&4
- HITACHI LED Display - Card 5
- Opto-Coupler Input/Output - Card 6
- pan/tilt Communications - Card 7
- Power Supply Unit (PSU) - Card 8.

The Controller Unit is powered from a single 240V power supply, from which the on-board PSU supplies all the voltage requirements for the cards, led, camera, motors, etc. The system is ventilated by a low voltage, four-inch diameter fan, based on the rear of the rack mounting. Input from the pan/tilt head is via the communications card.

Output from the Unit is via cards (1) and (2). The Euro-size pcb cards are mounted and held in place using quick release pins, mounted at each corner of the front plates. Each card has a red LED surface mounted on the front plate to

indicate that the card is operational. Each Euro-card fits into a standard back-plane in the sub-rack mounting using DIN 41494 Euro-size 32/64 way connectors.

4.3.3.1 The 2-Axis Stepper Motor Programmable Controller Board (Card 1)

The 2-axis stepper motor programmable controller board is designed to control one or two stepper motors via stepper motor drive cards. The board is programmed via an RS-232 serial link, with a suitable programming device. The board has the capability to move both axes at the same time, or each independently of the other.

The board has the capability to be “daisy-chained” with other processor boards, where one acts as a “master” board co-operating with the other “slave” boards. The combined effect of this is increased processing and control capabilities. The board has 32k x 8 bits of non-volatile RAM memory. Stand-alone operation is envisaged, hence we require a non-volatile memory, as power may be lost at any time.

The stepper motor controller board uses a 5V regulated dc supply (of less than one Amp), which is supplied by the on-board PSU. The processor board provides facilities for user input/output, via Darlington opto couplers. This facility has been included to allow up to eight inputs/outputs. These could take the form of a number of control functions, such as hand-shaking with a PLC, reading sensors or some other user defined interface. The processor board is fitted with a 64 way DIN 41494 connector, which fits into the back-plane of the Controller Unit.

The 4-phase hybrid stepper motors are driven by unipolar 2 Amp stepper motor drive boards, one for each stepper motor. These drive cards are located in cards 3 and 4 on the main Controller Unit. The drive cards convert the signals from the stepper motor processor controller board, into the required stepper motor sequence for the hybrid stepper motors. There are two modes of operation for energising the motors, via the driver cards. These are Full Step or Half Step

modes. To step a motor in a particular direction, a specific switching sequence for the drive transistors Q1-Q4 needs to be followed. If the sequence is followed, i.e., unipolar full step mode, it results in the rotor advancing through one complete step at a time.

4.3.3.2 Stepper Motor Drive Cards (Cards 3 and 4)

The 4-phase hybrid stepper motors are driven by unipolar 2 Amp stepper motor drive boards, one for each stepper motor. These drive cards are located in cards (3) and (4) on the main Controller Unit. The drive cards convert the signals from the stepper motor processor controller board, into the required stepper motor sequence for the hybrid stepper motors. There are two modes of operation for the stepper motor, Full Step or Half Step modes of operation.

4.3.3.3 HITACHI Alpha-Numeric Display (Card 5)

The HITACHI alphanumeric LED type display is provided so that process and control messages can be built into programs. This functionality will aid an operator and positional information on the pan/tilt head and other useful information, via the LED on the front of the Main Controller Unit. The messages are alphanumeric in format. This type of LED is an intelligent, alphanumeric, dot matrix module with integral CMOS controller and driver IC's. The HITACHI has a high contrast and a wide viewing angle, with LED back lighting. The LED display is incorporated into an in-house built Euro-size pcb, incorporating a DIN 41494 32 way connector, and modifications to enable correct voltages to be applied from the on-board pcb.

4.3.3.4 Opto Coupler Input and Output (Card 6)

Darlington I/O opto couplers have been installed on the Programmable Controller Board. These Input/Output devices can be programmed to drive other devices that may be added to the AVS.

4.3.3.5 Switch Mode Power Supply Unit (Card 8)

A 25-Watt, triple output, switch mode PSU was the basis for the on-board PSU based in the Controller Unit. The Unit will provide +5V and +/- 15V as output voltage. The PSU was incorporated into a Euro-size card and modifications made, to enable safety cutouts to be implemented. The PSU card incorporates an on-board fuse, and a surface mounted fuse on the front plate of the card. The PSU has a voltage regulator to ensure there is a smooth voltage supply. The voltage is supplied via a standard EEC lead. All electrical safety standards have been met, and the unit is fully PAT tested and correctly earthed.

4.3.4 Tractor Unit

For the purposes of evaluating the prototype system, the pan/tilt head and camera, together with a local power supply, were mounted on a small, 6-wheel, radio remotely controlled tractor, which could be placed in a laboratory-based NME sewer pipe (Figure 4.4).

Figure 4.4 . Active vision camera mounted on 6-wheel tractor.

4.4 AVS Software Architecture

The software required to implement the AVS active states (Figure 4.1) features two image analysis modules, 'Detect VP' and 'Detect Feature', and two camera control states, 'Fix VP' and 'Fix Feature'. The latter two make use of a single control module for the pan/tilt head and camera. The remainder of this section describes the development of the camera control software and then describes the image processing modules.

4.4.1 Control Module

A small library of control routines has been developed. To meet the requirements of the programmable controller hardware, the chosen programming language is RSL, a structured programming language very similar to 'C'. The programmable controller board has an enhanced EPROM and the control software can be installed in the EPROM to facilitate autonomous operation.

The control software to be used within the AVS takes account of mechanical and control constraints of the AVS.

4.4.1.1 Mechanical Constraints

Mechanical constraints arise from the hardware engineering of pan and tilt head. The hardware allows multiple turns ($n360^\circ$) of both the main body of the pan and tilt head, and of the attachment for the CCD camera head. This must be restricted via software design to limit the extent to which the pan/tilt head may rotate to avoid snaring of the communications cables.

Other mechanical constraints relate to the rotational velocities and accelerations that may be applied to both the main body of the pan/tilt head and the camera attachment. Software must control the momentum of each of these items in order to optimise positional integrity. Velocities and accelerations have been determined and these are incorporated into the software design. We maintain a

uniform angular velocity in each axis of rotation, though not necessarily the same velocity in each axis.

4.4.1.2 The Control Element of the AVS.

The kinematic control for the camera was developed using a LINKLAB, a software package designed to take a description of a robotic effector in terms of joints and link length and return code which is a kinematic controller for the effector. The AVS pan/tilt head and camera can be represented as three-link model (Figure 4.5). The controller computes the desired pan and tilt rotations for each of the two stepper motors in the pan/tilt head. The LINKLAB simulation environment allows verification of the controller's performance prior to implementation (Appendix 2). The controller software was implemented in RSL.

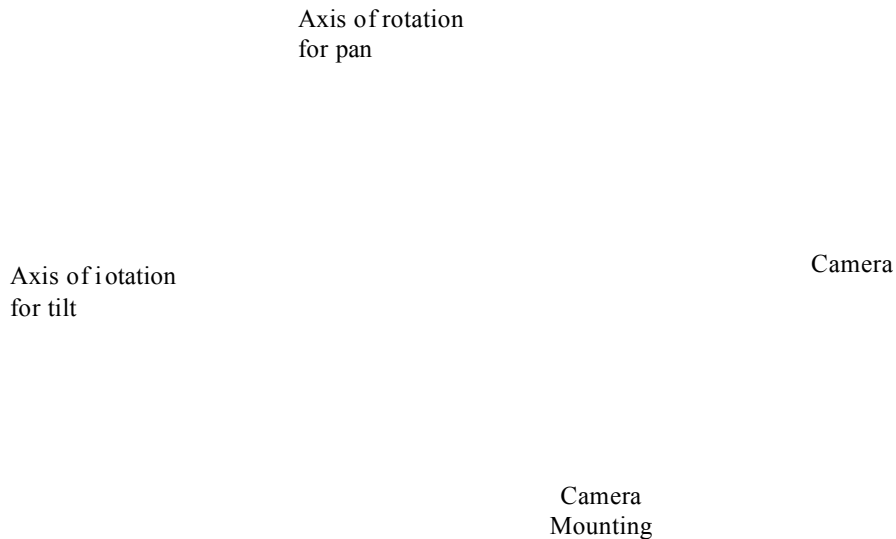


Figure 4.5 Wire-frame illustration of the pan/tilt head and camera illustrating links used by LINKLAB to determine a kinematic controller (see Appendix 2 for detail).

4.5 Image Analysis Software Modules

There are three image analysis states in the control statechart of Figure 4.1, Determine VP, Detect Feature and Analyse Feature. For the purposes of this feasibility study, the only features of interest were the VP and coaxial cracks. (The development and evaluation of the software to estimate VP, and to detect coaxial cracks form a central part of this feasibility study and the full details, together with results, are discussed in subsequent chapters.)

The software modules were developed within the TINA Vision Research Environment made available by the University of Sheffield Artificial Intelligence Vision Research Unit (AIVRU) .

4.6 Hardware and Software System Summary

The completed prototype system comprises the robot unit with pan/tilt head and CCD camera connected by a control and communications cable to an control unit. This, in turn, is connected to A PC, which has hardware and software for video image acquisition and analysis. Software on the PC is summarised in Table 4.2.

The evaluation of this system is described in Chapter 7 following description of the software modules for vanishing point detection and coaxial crack detection in Chapters 5 and 6, respectively.

Table 4.2 Software on the AVS PC.

MS - WINDOWS	LINUX 2.0
Video Blaster SE software	X – Windows
Terminal Emulation	GNU - C
CCD Feedback	XXGDB
AVS Software Interface to Programmable Controller	TINA Vision Research Environment
	LINKLAB Robotics Research Tool
	Image Processing Environment (XGRAB)

5 Detecting Vanishing Points in Images of Concrete Sewers

The VP of a sewer pipe can be exploited in at least three respects:

1. If the camera is oriented so that the VP lies on camera axis, then the VP is centred in the camera image. Accordingly, the best-illuminated and most highly resolved region of the image lies in the periphery of the image, as this is where the wall of the sewer is nearest to the camera. When examining the image for features of interest, an appreciable region in the centre of the image of the video stream can be excluded from processing because, features, especially cracks, are most reliably detected in the periphery of the image. This approach offers efficiency gains in image processing.
2. The VP can be used as a reference point for the geometry of the pipe. The diameter of a relatively undamaged clay or concrete sewer pipe is known at the outset of a survey. Thus, if the camera and pipe axes aligned, it is possible to estimate the dimensions and relative distances of features appearing on the wall of the pipe. The alignment of the axes is never perfect. Errors in estimates are proportional to displacement of the optical centre of the camera from the pipe axis. As long as this is small, especially in relation to the diameter of the pipe, then the errors will be small.
3. The VP may be used in controlling the path of the robot. As the tractor moves forward, the alignment of the camera axis and the pipe axis is displaced. If the camera has not moved (rotated) relative to the tractor upon which it is mounted, then the angular displacement of the of the pipe and camera axes may be used to derive a correctional steer for the tractor.

The vanishing point usually lies towards the centre of sewer survey images and may reasonably be expected to lie within the image boundary. Although the survey camera may yaw away from the axis of the pipe this rotation is unlikely to be severe enough to move the vanishing point outside the image frame. By assuming the vanishing point to be visible in the input image we avoid the need for both the probabilistic scheme of Lutton et al. [1994]. The major difficulty in locating the vanishing point in images of clay sewers is the lack of geometric

features; longitudinal lines are not marked on the wall of concrete and clay pipes. Intensity, rather than feature-based techniques must be used.

The following sections describe attempts to locate the vanishing point of a small-bore concrete sewer using two such intensity-based methods. The first method is that of Taylor et al. [1998] (see Chapter 3) and the second is an implementation of a new method based upon multiple luminance thresholds (as originally suggested in the summary of Chapter 3). These methods are evaluated using library images from a sewer survey conducted in Acton Town, London. The evaluation compares the relative performance (accuracy and speed) with which the methods make an estimate of the VP. The outcome of this evaluation is used to inform the laboratory-based experiments on the prototype AVS (Section 7).

5.1. Intensity-based Vanishing Point Detection

Consider a survey camera viewing the internal wall of a clay or concrete sewer illuminated by a single point light source. Light travelling from the pipe wall to the imaging plane of the camera decays with an inverse square law. In an ideal pipe, with the point light source and the principal axis of the camera coincident with the cylindrical axis of the pipe, the grey-level image of the pipe would appear as a series of concentric grey rings with grey level reducing towards the centre of the image. The vanishing point would lie at the centre of that pattern. Of course, the camera is not, in general, parallel to the pipe axis. (If it were, the VP would be fixed at the image centre and no new information would be provided.) And not even the most pristine sewer pipe is ideal.

In practice we see a roughly elliptical dark area near the middle of the image. The central portion of this area is usually black, producing too little light to register on the CCD array of the camera. Image intensity becomes steadily higher with distance from the centre of the region; producing a series of roughly concentric, approximately elliptical, “isoluminance” contours along which image intensity is constant. For a fixed light source, these contours are a function of the relative

orientation of the camera and pipe axes and can in principle be used to estimate the position of the vanishing point.

Taylor et al. [1998] use Gaussian smoothing to reduce noise before applying a single threshold to generate an “archipelago” of dark regions in the image. They assume that the largest such region includes the dark region generated by the most distant points in the pipe and that that the region is roughly symmetrical about the vanishing point. Finally, they compute the centroid of the dark region and take this as a point estimate of the vanishing point. This approach is computationally cheap but very sensitive to noise and any violation of the assumptions made. Broadhurst [2000] applied this method to pairs of images, captured from the same viewpoint but under different lighting conditions. The pair of VP estimates produced was then combined by simple averaging to produce a slightly more robust estimate. The intention in both cases, however, was only to locate a region of (dis)interest around the vanishing point (option 1 above). In these circumstances accurate estimation of the vanishing point is not required.

Tsukiyama [1995] describes a multi-threshold method designed to solve a similar problem, showing that the relative orientation of two planar surfaces can be estimated from the pattern of elliptical isoluminance contours that arise when both surfaces are illuminated by a single, point light source. This pattern is extracted by applying a set of thresholds to identify isoluminance contours, segmenting each contour into circular arcs and using a least-squares method to estimate the centre of the pattern from the set of circle centres. This algorithm is similar in structure to that of Cooper et al. [1998] who locate the vanishing point in images of brick sewers, and raises the possibility of using multiple thresholds, contour fitting and a least squares approach to produce a vanishing point estimate that is reliable enough to allow both the determination of regions of interest and some degree of robot/camera control (option 3 above).

5.2 Vanishing Point Estimation from Isoluminance Contours

Estimation of the position of the vanishing point of a concrete sewer proceeds as follows:

1. Apply Gaussian smoothing to reduce noise. The Gaussian parameter is typically in the range 1.0-3.0.
2. Apply a series of image thresholds, again determined empirically. This generates contours that segment the image into bands of similar grey level. Under ideal conditions (above), the boundaries of these bands, the isoluminance contours, would be circular. Under non-ideal conditions, they may be described as conic sections. (c.f. Pan et al. [1994, 1995]).
3. Segment and fit conic sections to the isoluminance contours. This is achieved using techniques incorporated in the TINA environment and based on Pridmore et al. [1987].
4. Estimate the vanishing point position by taking the mean of the centres of the conic sections.

The advantage of the conic section method is that should be more robust than the single threshold technique and gives an estimate of precision. The disadvantage is that it is a more computationally expensive method.

5.3 Evaluation

Digitised images of from the Acton Town sewer pipe video surveys have been captured. The evaluation of the accuracy of the two methods is necessarily subjective in that the true VP in the images cannot be known. The protocol is illustrated below.

To illustrate the evaluation, four pairs of images are presented below (Figure 5.1(a-d)). Each image is analysed by each of the two methods presented using the same computer hardware (see above). The cross hair in the image represents the VP as estimated by the respective method. The method of Taylor et al. is labelled as the 'Region Method' and the method using isoluminance contours is labelled the 'Conic Section Method'. In the images produced using the Region Method, the diameter of the circle about the estimated VP corresponds to the description of the neighbourhood of the VP as described in Chapter 3. The circle about the estimated VP in the Conic Section images is set at an arbitrary fixed value and has no significance.

Considering each image in turn:

Image 1: This is a reasonably unproblematic image and both methods might be thought to produce reasonable estimates. Closer inspection suggests that the Conic Section method is marginally better as it is more centred in relation to the 2 or 3 contours associated with pipe joints. The centroid computed by the region method responds to the irregular (keyhole) shape of the thresholded dark region.

Image 2: This is a much more difficult image to analyse. Again, both methods achieve reasonable estimates. The conic section method produces the marginally better estimate, as the crosshairs appear to be inside the faint image of the most distant pipe joint. The diameter of the neighbourhood computed for by the region method includes much of the poorly illuminated wall of the pipe on the right of the image and so the VP estimate will have been generated using pixels not associated with the VP itself.

Image 3: In image 3, the region method has identified a wholly inappropriate region as the basis for its estimate. The conic section method is robust in the face of this problem.

Image 4: Again, the region method has identified an inappropriate region as the basis of its estimate. The conic section method is considerably better, appearing, as it does, to have clearly picked up the edges associated with the clearly visible sequence of pipe joint flanges.

Image 1: Conic Section Method

Image 1: Region Method

(a)

Image 2: Conic Section Method

Image 2: Region Method

(b)

Image 3: Conic Section Method

Image 3: Region Method

(c)

Image 4: Conic Section Method

Image 4: Region Method

(d)

Figure 5.1(a-d). Comparison of two methods for estimating the VP of sewer pipe images. VP estimates indicated by red cross hairs.

5.4 Discussion of Results

In each case, the Conic Section Method gives a better estimate of the VP.

Inspection of images 3 and 4 revealed the fundamental weakness of the Region Method: the single threshold to define the largest connected 'dark region' has included appreciable regions of the image which are not associated with the VP.

The effectiveness of the conic section method, and especially so in image 4, suggests that the method could usefully be compared to the method of Xu at al [1998] used to detect pipe junctions at some later date.

The region method, though seemingly the less accurate of the two methods, is very much quicker in terms of processing time. For practical, real-time, applications, we have to trade-off computational efficiency for accuracy/precision. In this case we are dealing with processing times that generally differ by roughly $(n \cdot t_c)$ where n is the number of contours fitted and t_c is the time to fit the centre of one conic section. In this evaluation, t_c was about ten minutes! Roughly, ten times the time for the Region-based method. Although the significance of this will diminish with improvements in computing hardware, at this time, the use of the multiple contour (conic section) method is not practicable.

6 Crack Detection

The prototype AVS must detect features of interest within the inner surface of sewer pipes. This feasibility study is concerned specifically with the detection of cracks in the sewer wall. As described in Section 5, using the estimated VP as a reference it is possible, to a reasonable approximation, to align the camera axis with the cylindrical axis of the pipe. When this is done, the best illuminated and most highly resolved region of the image lies in the periphery of the image as this is where the wall of the sewer is closest to the camera. <

As the survey tractor advances through the pipe, sampling and examining images captured from the video stream, candidate cracks will migrate into the periphery of the image from the centre of the image. Of course, there may be more than one candidate crack in any image (Figure 6.1).

Figure 6.1 Multiple Cracks near a pipe junction in NME Sewer Pipe (Acton Survey).

The primary objective of the work described in this chapter was to operationalise the 'Detect Feature' state (Figure 4.1) so that the system carries out feature specific processing of an image. This involves using image processing modules,

developed within the TINA environment [TINA], so that they could be used in this application context and includes determination of values for image processing parameters that are effective at highlighting cracks in the sewer wall. A second objective was to develop an approach to determining a region of interest, associated with a candidate crack, to be fixed (centred) by the AVS prior to detailed feature analysis. Again, use of library images from previous video surveys means that this is a subjective exercise. As in a conventional CCTV survey, judgement plays a significant part in determining which candidate cracks are of interest and which are not. The extent to which both of these objectives were met was later evaluated in the laboratory prototype (Section 7).

In order to achieve these goals, an image analysis module, CrackTool, was developed and added to the library modules of the TINA vision research environment. In processing an image, CrackTool first applies Canny edge detection and then processes the resulting image looking for edges that may be associated with cracks (see 6.2). Given a candidate crack, CrackTool then uses a simple heuristic to compute a 'feature centre' which is an estimate of the centre of the region of interest. If there is more than one candidate crack in the image, a series of such centres is generated. Each of these steps is now described.

6.1 Edge Detection

Many edge detection algorithms or operators exist; most are based on image differentiation. The first derivative of image intensity is usually estimated and significant peaks, which correspond to edge locations, are marked. In 1986, Canny proposed an operator, which has become the de facto benchmark method for edge detection [Canny, 1986]. This operator is derived from an optimisation procedure seeking good detection (an edge detection algorithm should mark all edges and only edges), good localisation (points located should closely conform to the corresponding edge) and minimal response (each edge should be uniquely marked).

The Canny algorithm involves removal of image noise by Gaussian smoothing followed by differentiation and a search for significant peaks. Canny edge detection requires a number of parameters to be specified: (Gaussian) smoothing (σ), string length (l) and thresholds (t_{\min} , t_{\max}). A relatively low σ value provides minimal smoothing; maintaining sharp boundary detail but permitting a high volume of edge data, some of which may be the result of noise. A relatively high σ value gives greater noise reduction but will blur edge boundaries and may lose significant features. The line length threshold enables short edge strings to be discarded, discounting unwanted edge detail. Illumination or other effects may however cause significant line structures to be fragmented. High length thresholds will subsequently discard these fragments. The thresholds, t_{\max} and t_{\min} , are used to assess the height, and therefore significance, of peaks in the first derivative of intensity. The determination of appropriate values for these parameters results in an improved the signal to noise ratio in the images.

If a specific feature of known shape and dimensions is to be recovered from an image, it is possible to use an automated thresholding technique to vary parameter values over a predetermined range so that the routine gives the 'best' image (e.g., based on uniformity and shape measures [Sahoo et al, 1988]). However, cracks do not meet the criteria of a 'known shape' and no such reference feature is shared in all sewer images. Accordingly, the task of choosing parameter values which provide good results across a range of actual survey images is a subjective task.

In order to determine a set of parameter values that give satisfactory results on actual surveys I carried out a study using images from the library video footage of the Acton Town Sewer Pipe Inspection Survey. A total of eight images were selected from a sub-set of approximately thirty original images acquired. These eight were identified as giving a fair representation of the range of crack features found commonly within sewer pipes. Each of the eight images was treated as follows. The original digitised image captured from the video stream using hardware and software described earlier (Ch.4) was subjected to Canny edge detection as the Canny parameters were varied over a range of values.

Appropriate threshold values were quickly determined as $t_{min}= 5$; $t_{max}= 10$. The procedure to identify appropriate values for other parameters can be illustrated using an example, as follows.

Figure 6.2 presents a sequence of 12 Canny images for one of the eight original images from the Acton Town Survey sequence. Figure 6.2(a) shows the original sampled image: a pronounced coaxial crack can be seen in the top right quadrant. Figure 6.2(b) is the same image with the graphical output of the Canny Edge detection imposed upon it ($cr=1$, $t_{min}=3$, $t_{max}=5$, $I=5$). The label (index) assigned to each processed image, together with the values of the Canny parameters that generate that image, are given at the foot of each image.

6.2(a) Crack 1 (Canny processing not applied)

6.1(b) C2rl 01: ($G \rightarrow$, t_{min}-3, t_{max}~5, l-5)

C\ CL-ut \rightarrow V\A S V A S V ^ 1 ^ $\&$
 \wedge / i v v % ^ r - (C
 r \ v K ~ r ^ r v " ^ / A - ! * " >
 x C 2 / \ }
 Y u * ^ , / L y s .
 a L ^ . j u y y * f . l - u t v i
 * > \) * r > . ^ s t y g g S . 5 A - ' ' n ^
 < t & , > c \ v * a
 ^ a V M V O ^ / i) 4 j
 f a C / L o T ^ ^ > ' *

6.2(c) crl elm: ($\sigma \rightarrow$, t_{un}_3/ t_{max}-5/ l-5)

v

6.2(f) `crl_e4`: (`<7=1`, `tmin=3`, `tmax=5`, `l=20`)

■v

6.2(g) `crl_e5`: (`<7=1`, `tmin=3`, `tmax=5`, `l=30`)

>X

& // //
• * > //

/ ■ F /

6.2(h) cr1_e6: (<7=2, tmin=3, tmax=5, 1=5)

\\

J

6.2(i) cr1_e7 -4, tmin=3 / tmax=5, 2=5

6.2(j) $crl_e8: \{ (7=5, \text{tmin}=3 / \text{tmax}=5, 1=5) \}$

| V |

6.2(k) $crl_e9 \quad -2, \text{tmin}-3 / \text{tmax}-5, 1-10$

6.2(1) `cr1_e10`: ($\leq 7=4$, $t_{\min}=3$, $t_{\max}=5$, $l=10$)

6.2(m) `cr1_e11`: ($\leq 7=5$, $t_{\min}=3$, $t_{\max}=5$, $l=10$)

Figure 6.2. Canny Edge detection sequence applied to image of 'Crack 1' for different Canny parameter values. (a) Original image from the Acton Town sequence. Note the coaxial crack in the upper right quadrant. (b) Edge detection applied to the upper right region and superimposed on the original image. (c)-(m) remaining Canny images in the search.

For the sequence in Figure 6.2, a subjective assessment suggests that the main feature of interest (the appreciable coaxial crack) is distinguished from background in images 6.2(i) and 6.2(l) and thus, that parameter values of: $\sigma=4$; $t_{\min}=3$; $t_{\max}=5$; $5 \leq l \leq 10$. A similar judgement was made over all 96 images examined. The results obtained suggest that effective parameter values are $\sigma = 4$, $l = 5$, lower and upper thresholds of 3 and 5 respectively.

This subjective, empirical determination of parameters is common in image analysis and is appropriate as there is good reason to believe that the parameters chosen in this way will be effective on a range of sewer images. One of the most attractive practical attributes of the Canny algorithm is the stability of the thresholding with hysteresis method. As t_{\min} and t_{\max} determine an allowable range of 1st derivative values, rather than a single hard threshold value, it is comparatively easy to determine a range of threshold values that are applicable across similar images. The remaining parameters, σ and l , effectively determine the width and length, respectively, of the features detected. The apparent size of a crack in any given image depends upon the actual parameters of the feature and the viewpoint - particularly viewing distance. Seeking cracks in the periphery of the image restricts the effective viewing distance, and so puts a constraint on the apparent width of any cracks. This in turn restricts the range of σ values required to detect them; the σ determined for a small set of images should be valid in general. The length threshold only seeks to reject edge strings that are too short to arise from cracks. Restricting viewing distance by only searching the periphery similarly eases selection of l as viewing distance only varies a little. The apparent length of the shortest edge worthy of consideration will also vary only slightly. A

value chosen to suit a small test set can reasonably be expected to be applicable to a wider class of images.

6.2 Coaxial Cracks and Crack Centres

Having determined edge detection parameters that gave subjectively good results in highlighting cracks of interest, the next objective was to identify an image region containing the crack of interest with the intention of orienting the camera towards these cracks for detailed inspection/analysis.

The features of interest for this feasibility study are coaxial cracks. Code was developed which processed a Canny edge image to identify this type of crack (Appendix 3). Given a set of edges identified by Canny edge detection, as above, the algorithm to distinguish coaxial cracks from other features and noise applies a sequence of criteria which progressively eliminate those features to be excluded from consideration. In order to visualise this processing, the edges eliminated at each stage of processing are assigned different colours.

The Canny operator as implemented in the TINA environment represents edges in any image as a linked list of edge points. An image is represented as a list of such linked lists. This data structure provides the input to the crack detection routine (CrackTool). A two-stage 'filter' determines candidate cracks. First, a set of tests is applied to each point in each edge in an image. All edges which survive this first filter are then considered pair-wise. Each pair which passes the second 'filter' is considered to be a candidate crack. Before the tests were applied each string was Gaussian smoothed to reduce noise. The Gaussian smoothing was applied independently to x and y coordinates, both expressed as functions of arc length.

The first filter tests applied to each edge are as follows:

- 1) Exclude edges that are not sufficiently in the periphery of the image. A rectangular boundary, centred on the vanishing point, defines the region of

interest. If the percentage of points in an edge that fall inside the rectangle exceeds a user-defined threshold, that string was rejected (coloured blue on the output image).

- 2) Exclude edges that are too short to be cracks. Remaining edges whose length is below a user-defined length threshold (i.e. which were largely outside the central region but too short to be considered further) were rejected (coloured green). In practice this threshold was usually set to the same value as the Canny length threshold.
- 3) Reject edges that are not (approximately) co-axial. For each remaining edge, a straight line is fitted through the end-points of that edge. The length of the normal (perpendicular distance) from the vanishing point to each such line is computed. If that distance exceeds a user-defined threshold the line is rejected (coloured yellow).
- 4) Remaining edges (coloured red on the output image) are passed to the pair-wise test.

The second filter is applied as follows:

- 1) Determine what proportion of points in a given edge lies within a threshold distance of another edge. To test edges A and B, each point in A is considered in turn. The distance between this point and each in point in B is measured and the minimum such distance for that point is found. A count is kept of the number of points in edge A whose shortest distance to edge B is below a threshold value.
- 2) Select candidate cracks as pairs of edges. If the percentage of points in A that are within a minimum distance to B is above a user-determined threshold, the edge pair is considered to be sufficiently similarly oriented and closely spaced to be a candidate crack (coloured cyan on the output image).

Finally, an estimate of the centre of interest for each crack is determined. The estimate is computed as the centroid of the four end-points of the two matched edges. (Graphically, the centre is represented as a cyan cross surrounded by a cyan circle.) The result of applying the CrackTool analysis to an image is illustrated in Figure 6.3: the edges are coded as above; crack centres are highlighted as Cyan cross hairs.

Consider Figure 6.3(a) which is the 'Crack 1' image in Figure 6.2. The tool appears effective in detecting the coaxial crack in the upper right quadrant of this figure (Figure 6.3(b)). The blue crack edge(s) not in the periphery (furthest from the camera) are poorly resolved, being made thin by perspective. The yellow edges are not coaxial. The rightmost yellow edge might be considered coaxial for much of its length but the heuristic test for orientation fits a line through the end points and this line is clearly not axial. (A least-squares line fit over all points in the edge could be considered as an alternative approach at a later date.) The red edge has no matching edge; the cyan edges seem effective in defining the clearly resolved crack of interest. The multiplicity of cyan cross hairs indicates that the processing has identified a number of edge pairs, each of which has an associated centre (although it is not possible to distinguish them in the graphical image presented).

(a)

(b)

Figure 6.3 (a-b) CrackTool highlights edges: blue edges are not sufficiently in the periphery; green edges are too short to be cracks; yellow edges are not sufficiently axial in their orientation; red edges have no matching edge to form a crack; cyan edge pairs are candidate cracks: centres are highlighted as cyan cross hairs.

Figure 6.4 provides two further examples of images processed using the crack detection software module, CrackTool.

In Figure 6.4(a) three cracks (and respective centres) have been detected. The uppermost and lowermost of these are very short and must be considered false positives, suggesting that an adjusted length threshold may be warranted. However, the third crack detected is the peripheral component of a coaxial crack that extends into the pipe and this may not have been detected if the length threshold had been greater. Much the same can be said of Figure 6.4(b). However, in this case, the two uppermost cracks appear to be part of a crack across the roof of the pipe that does not, as a single feature, fall within the criteria which select for coaxial orientation.

6.3 Summary

This chapter has used library images of sewer surveys to establish the feasibility of detecting coaxial cracks in the periphery of images. Having determined a set of parameters for Canny edge detection, it described the development of an image analysis tool, CrackTool, which seeks to identify candidate cracks.

For the purposes of the prototyping exercise, these results suggest that the crack detection software does, indeed, select candidate cracks. However, within the features detected, there are both false-positives and false-negatives. It will not be possible to eliminate both of these forms of error and, in practice, a balance will have to be established that trades-off wasted analysis and reporting of false-positives with failure to report false-negatives. The determination of the balance point will depend upon field trials of any advanced prototype and is beyond the scope of this initial study.

7 Laboratory-based Prototype Testing

The final piece of work undertaken in this feasibility study was to test the prototype tractor-mounted AVS in a laboratory setting. The goal was to demonstrate that the software developed for the prototype using the Acton Town Survey images and the camera control software (developed using LINKLAB) worked effectively in a test environment. This chapter describes the laboratory-based test rig used for the prototype testing; it illustrates the application of the image analysis routines to images from the test rig; finally, it illustrates the operation of the AVS to control camera head movement.

7.1 Test Rig

The tractor with pan/tilt head and camera is illustrated in Figure 7.1. In figure 7.2, the tractor is positioned in the mouth of the concrete sewer pipe. The far end of the pipe is masked to eliminate light entry. Coaxial cracks are simulated using black tape. The tractor is connected via a communications cable to the system control unit. In turn, this is connected to a PC which runs the image capture and analysis software (see Table 4.2, above).

(a)

(b)

Figure 7.1 Prototype AVS. (a) radio controlled tractor mounted AVS. (b) The pan/tilt head and camera and single, fixed-axis, light source as mounted at the front of tractor.

Figure 7.2 Prototype AVS in the mouth of a concrete NME sewer pipe. The tape used to simulate coaxial cracks is also visible, running into the pipe at about 2 o'clock and 9 o'clock.

7.2 Image Analysis Tests

With the robot positioned in the test rig (Figure 7.2) pipe images can be captured and then processed using the TINA modules described in Ch.5 (VP determination - region method) and Ch.6 (crack detection).

7.2.1 Initial Qualitative Evaluation

The images of Figure 7.3 show a selection of the pipe images that were used to test the modules. In each image, the direction of the camera axis is indicated by white cross-hairs. The VP is located using the method of Taylor et al [1998] as evaluated in Chapter 5. The estimate of the VP in the image is indicated by blue cross-hairs. Finally, the coaxial cracks detected by the CrackTool module as developed in Chapter 6 are defined by the cyan edges and the 'crack centres' are marked by cyan cross hairs.

(a)

(b)

(c)

Figure 7.3. Processed images from test rig. The images (a-c) provide appropriate qualitative evidence of the effectiveness of the state-related analysis routines (white cross-hairs indicate direction of camera axis; blue cross-hairs the estimated VP; cyan cross-hairs the crack 'centres').

These images (and others not illustrated here) provide sound qualitative confirmation of the effectiveness of the VP determination and crack detection modules. The 'ideal' conditions of an unused pipe significantly diminish the problem of noise in each image. The VP determination process has, perhaps, benefited from this in particular.

7.2.2 Quantitative Evaluation for VP Estimate

Table 7.1 presents quantitative data to support the initial qualitative test just described. Images were captured with the camera axis oriented away from the VP. The captured images were represented to the TINA VP location module such that the VP was subjectively located approximately at $x = y = 150$ and the direction of the camera axis was 'off centre' in respect of the VP. The TINA VP module then computed the estimated VP position.

Table 7.1. Computation of estimated coordinates of VP (in pixels). The position of the camera axis is given by Camera-x and Camera-y, respectively. The estimated VP is given by the third and fourth columns. The actual VP was (subjectively) centred at 150:150.

Image	Camera-x	Camera-y	estVP-x	estVP-y
1	231	38	158	146
2	93	41	157	145
3	249	237	157	147
4	68	237	158	145
5	130	77	157	147
6	100	76	157	145

The computed estimates of the VP in Table 7.1 show good precision. In respect of accuracy, given that the original centring of the VP was subjective, it is not possible to say whether the deviation in the estimates (in relation to the subjective target 150:150) arises from a lack of accuracy or from some systematic error. If so, the error in accuracy is still marginal and well within acceptable bounds for all practical purposes.

7.3 AVS Control

Figure 7.2 (above) shows the robot positioned in the test rig. Figure 7.4 shows similar images taken from closer in (the pipe joint flange forms an arc across the top of the image). The tape which simulates the coaxial crack is at about 8 o'clock. In Figure 7.4(a) the camera is not oriented to any feature. In Figure 7.4(b) it has fixed the VP. In Figure 7.4(c) it has centred the region of interest containing the crack. Testing of the AVS control system has two elements described, in turn, below.

(a)

(b)

(c)

Figure 7.4. Active control of the camera head. (Note: the bright spot in the centre of the image is the reflector of the fixed-point illumination source on the tractor.)

7.3.1 Recovery of VP alignment

In the first, the camera axis was aligned with the pipe VP of fourteen images and then displaced. The aim was to see if the system could recover the original position by estimating the pan and tilt angles needed to re-centre the VP. The effects of the displacements are illustrated by Figures 7.5(a-b). The test results are given in Table 7.2. The first column identifies the test image (tif file); the second and third columns are the introduced displacements (expressed in stepper motor steps - half step mode); the fourth and fifth columns are the calculated

angular displacements (expressed in stepper motor steps - half step mode)
needed to recover the estimated VP as computed by the pan-tilt software
controller.

(b)

Figure 7.5. Effect of introducing controlled displacements away from the VP into
direction camera axis.

Table 7.2. Displacement and recovery of position in stepper motor half-steps (1 half-step = 0.9 degrees). The signs differ because recovery of the VP requires the reversal of the introduced displacement.

Image (*.tif)	Disp. Pan	Disp. Tilt	Calc. Pan	Calc. Tilt
t1	-	18	-	-18.5656
t2	-	9	-	-9.6090
t4	-	-18	-	18.1428
t5	18	-	-18.2671	-
t11	9	-	-10.2325	-
t6	-18	-	18.0000	-
t7	-18	-18	18.4002	18.2839
t8	18	18	-18.0002	-17.5758
t9	-18	18	18.2671	-18.1427
t10	18	-18	-19.5890	17.1482
t12	9	9	-10.2325	-9.6090
t13	-9	-9	10.2325	9.6090
t14	-16	-13	16.7894	12.9187
t15	-13	-16	13.2065	16.4312
t16	-13	-13	13.2065	12.9187
t17	-16	-16	17.1948	15.9985

Over the 16 images tested the average pan error in the was 0.66 half-steps (SD=0.56) and the average tilt error was 0.37 half-steps (SD=0.27). These figures suggest good accuracy and represent good precision for the camera head controller.

7.4 Orient Camera Axis to Crack Centre

The final test carried out in the laboratory test rig was to confirm the ability of the camera to fix (orient to) an estimated crack centre (prior to detailed feature analysis). The sequence of images in Figure 7.6 illustrate the iterative orientation to features of interest. Figure 7.6(a) shows the original image in which two peripheral cracks are detected and their centres estimated. Given more than one centre, the AVS estimates a centre of centres and orients to the camera accordingly (Figure 7.6(b)). The process is now repeated (Figure 7.6(b)). A third iteration results in Figure 7.6(c). (Note that when orienting the camera to the region of interest, it is no longer appropriate to exclude crack analysis of the central part of the image.)

(a)

(b)

(c)

Figure 7.6. Iterative orientation of camera axis with regions of interest associated with detected cracks.

The results of this and similar tests provide evidence that the camera is able to orient to the region of interest associated with a detected crack. Although not of direct interest here, the problem of false-positive and false-negative cracks (anticipated in Chapter 6) is in evidence here. The images of Figures 7.5(b) and

7.5(c) illustrate that even in the ideal environment of the test rig, the crack detection tool highlights false positive features (see crack in top centre of each image respectively).

7.5 Summary

The laboratory-based tests for the AVS prototype have been presented. Using images from the test rig, the modules to estimate the VP and detect and compute centres for cracks have been tested, as has the software controller for the AVS camera control system.

In the somewhat ideal environment of the laboratory, the region method for VP estimation and the crack detection software have both performed more than satisfactorily. The AVS control (camera orientation) software has also proven capable of orienting the camera so that the camera axis is directed at the VP of the pipe as required.

8 Summary and Conclusion

This aim of this project has been to begin the development of a prototype robotic AVS that could be deployed for autonomous sewer survey.

Within the overall aim, the scope of the project had specific practical objectives (originally stated in Chapter 2):

- ii. identify and evaluate algorithms for initial detection of cracks/fractures in small-bore pipe walls given a single image of the pipe;
- iii. determine the camera kinematics required for feature fixation;
- iv. develop the system hardware;
- v. develop the system software;
- vi. test the prototype system on simulated faults in a laboratory environment;

Each of these objectives is considered below.

The first objective gave rise to related objectives once the VP was identified as a useful reference point in a sewer pipe image. If centred in the image, the VP allows us to selectively analyse the image periphery for features of interest (coaxial cracks). Two methods of VP estimation were evaluated. The first region-based method [Taylor et al, 1998] computes the centroid of a dark region generated by a single threshold. The second method, developed as part of this project, fits conic sections to contours generated by multiple thresholds and computes the mean centre of these. The accuracy of the conic section method was better than that of the region method but the time required to arrive at this improved estimate was an order of magnitude greater and was not appropriate for practical purposes given current PC-based hardware¹.

¹ This can be kept under review in light of continually increasing computer performance. Indeed, as this report is submitted, it is likely to be the case that the real time performance of both approaches has improved by at least an order of magnitude. This is probably insufficient to warrant a change in choice at this time.

A crack detection module was developed within the TINA environment. The performance of the crack detection tool is mainly influenced by the choice of edge detection parameters. Values for these were determined experimentally. These values may be adjusted on the basis of further experience, perhaps in a wider range of environments. The crack detection tool successfully detected cracks in both library images and on a laboratory pipe test rig.

The crack detection method described here took a comparatively simple, structural pattern recognition approach. Heuristic rules are written which describe the features expected of the image of a crack in the pipe wall. It might be argued that statistical pattern recognition, in which models are built from training sets of sample data, is a much more robust approach. In many situations this is the case. However, cracks in sewer walls vary considerably and it is not clear that a reliable statistical model could be built. At the very least, an extremely large training set would be required. The active vision approach taken here sought to avoid this problem by ensuring that inspection images were acquired under circumstances which allowed comparatively simple, structural methods to perform adequately.

A prototype tractor-mounted AVS has been developed. A radio-controlled tractor carries controllable pan/tilt head and camera. Control hardware and software for the head and for image processing and analysis has been developed and tested in a laboratory context.

In Chapter 4 a state-chart has been used to represent the important elements of an active process-based AVS (Figure 8.1). In considering the feasibility of continuing work in this area the outcomes of the project can be related to the statechart.

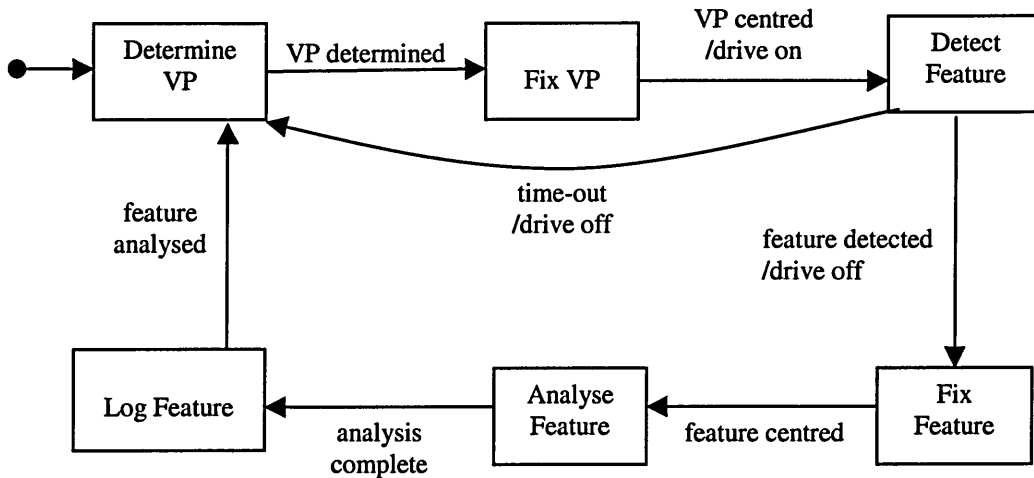


Figure 8.1. Statechart specifying system-level control of active vision system. Each state is an active state, associated with the performance of a system operation. (Reproduced from Figure 4.1.)

- ‘Determine VP’: Two, alternative, methods for implementation of the ‘Determine VP’ state have been developed and compared (Chapter 5).
- ‘Fix VP’: Software capable of orienting the camera so that the VP is ‘fixed’ (centred) in the camera image has been developed (Chapter 4) and tested successfully on library images (Chapter 5) and in a laboratory setting (Chapter 7).
- ‘Detect Feature’: Software has been developed which is capable of detecting candidate coaxial cracks in the periphery of both library images (Chapter 6) and in a laboratory setting (Chapter 7).
- ‘Fix Feature’: this state uses the same camera control software as ‘Fix VP’. The fixation point is the crack centre as estimated by the ‘Detect Feature’ state.

At this stage, there would seem to be evidence to suggest that further work could be undertaken to develop at least semi-autonomous survey technology.

Wirahadikusumah et al [1998] have set the broad goals for researchers in this area:

- more accurate and dependable survey information
- improved survey efficiency and economy
- reduced disruption from survey
- ease of deployment of surveying technology.

In respect of the first two of these goals, I suggest that the AVS control and analysis routines developed and tested for this project make a contribution in that they are able to obtain machine analysable representations of important features of interest. The ability of the AVS to identify and orient to features of interest reduces current demands upon human operators to do likewise.

The remaining two goals will depend upon the extent to which a robust and largely autonomous robotic platform can be developed upon which to deploy the vision technology described in this thesis.

References.

[Abbott and Ahuja, 1992] A.L. Abbott , N. Ahuja, The University of Illinois Active Vision System. Proc. SPIE Conf. on Applications of Artificial Intelligence XI: Algorithms, Techniques and Active Vision, Boston, November 1992, 757-768.

[Abraham et al., 1997] D.M. Abraham, T. Isley, R.K. Prasanth, R. Wirahadikusumah, Integrating sensing technologies for underground utility assessment. ASCE Conference on Infrastructure Condition Assessment, Boston, MA, August 1997

[Allen, 1989] P.K. Allen, Real-time motion tracking using spatio-temporal filters. Proc. DARPA Image understanding workshop, 1989, 695-701

[Barnard, 1982] Barnard, S.T., Models for interpreting perspective images, Proc. Image Understanding Workshop, Palo Alto, California, (1982), 193-202.

[Beardsley and Murray, 1992] Beardsley, P., Murray, D., Camera calibration using vanishing points. Proc. British Machine Vision Conference, (1992), 416-424.

[Broadhurst, 2000] S. Broadhurst, Non-Man-Entry Sewer Renovation Robot Characteristics. PhD Thesis, Sheffield Hallam University, Sheffield UK. 2000

[Brunnstrom, 1996] K. Brunnstrom, J. Eklundh, T. Uhlin, Active Fixation for Scene Exploration. International Journal of Computer Vision, 17 (1996) 137-162

[Campbell et al., 1995] G.Campbell, K. Rodgers, J. Gilbert, PIRAT - A system for quantitative sewer assessment. International No-Dig '95, Dresden Germany, September (1995) 455-462

[Canny, 1986] J. Canny, A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 8 (1986): 676-698

[Cooper et al., 1998] D. Cooper, T.P. Pridmore, N. Taylor. Towards the recovery of extrinsic camera parameters from video records of sewer surveys. *Machine Vision and Applications*. (1998) 11: 53-63

[Duda and Hart, 1972] Duda, R., Hart, P., Use of the hough transform to detect lines and curves in pictures. *CACM*, 15, (1972), 11-15.

[Fischler et al., 1982] Fischler, M.A., Barnard, S.T., Bolles, R.C., Lowry, M., Quam, L., Smith G., Witkin, A., Modelling and using physical constraints in scene analysis. *Proceedings of AAAI-82*, (1982) 30-35.

[Henry and Luxmoore, 1996] R. Henry, A.R. Luxmoore, A pipe-profiling adapter for CCTV inspection cameras: development of a pipe-profiling instrument. *Meas. Sci. Technol.*, 7 (1996) 495-504

[Kolesnik, 2000] Kolesnik, M. On Vision-Based Orientation Method of a Robot Head in a Dark Cylindrical Pipe. *Sofsem 2000---Theory and Practice of Informatics. Lecture Notes in Computer Science, Volume 1963, 2000*, pages 364-372

[Kolesnic, 2002] Kolesnik, M. Visual orientation in the sewer - adaptation to the environment. *Proceedings of ICPR'2002 - International Conference on Pattern Recognition. Quebec City, Canada, 11-15 August, 2002*.

[Kolesnik and Baratoff, 2000] Kolesnik, M., Baratoff, G. 3-D Interpretation of Sewer Circular Structures. *Proceedings of the Int. Conference on Robotics and Automation (ICRA'2000), 22-28 April, San-Francisco, 2000, Volume 2*, pages 1453-1458

[Kolesnik and Streich, 2002] Kolesnik, M., Streich, H. Visual Orientation and Motion Control of MAKRO- Adaptation to the Sewer Environment . In: B. Hallam, D. Floreano, J. Hallam, and J.-A. Meyer (eds.). From animals to animats 7. Proc. of the 7th Int. Conf. on Simulation of Adaptive Behavior. Cambridge, Mass.: MIT Press, 62-69. Edinburg, UK, 4-9 August, 2002

[Kuntze et al., 1995] H.B.Kuntze, D. Smidt, H. Haffner, M. Loh, KARO - A flexible robot for smart sensor-based sewer inspection. International No-Dig '95, Dresden Germany, September (1995) 367-374

[Lutton et al., 1994] Lutton, E., Maitre, H., Lopez-Krahe, J., Contribution to the determination of vanishing points using the Hough transform. IEEE Trans. PAMI, 16, 4, (1994), 430-435.

[Malik, 1987] 3e J. Malik, Interpreting line drawings of curved objects. International Journal of Computer Vision, 1 (1987) 73-104

[Moselhi and Shehab-Eldeen, 1999] O. Moselhi, T. Shehab-Eldeen, Automated detection of surface defects in sewer and water pipes. Automation in Construction 8 (1999) 581-588.

[Murray et al., 1995] D.W. Murray, K.J. Bradshaw, P.F. McLauchlan, I.D. Reid, P.M. Sharkey, Driving saccade to pursuit using image motion. International Journal of Computer Vision, 16 (1995) 205-228

[NSF, 1993] National Science Foundation, Civil Infrastructure Systems Research: Strategic Issues, A Report of the Civil Infrastructure Systems Task Group. January 1993.

[NWC, 1977] Department of the Environment/National Water Council (1977). Sewers and Water Mains - A National Assessment. NWC/DoE Standing Technical Committee Report No. 4 NWC.

[Paletta et al., 1999] L. Paletta, E. Rome, A. Pinz, Visual object detection for autonomous sewer robots. In IROS'99. Proc of the 1999 IEEE/RSJ International Conference on Intelligent Robots and Systems, Kyongju, South Korea, October 17-21, pp1087-1093

[Pan et al., 1994] X.Pan, T.A.Clarke, T.J.Ellis, The detection and tracking of Pipe Joints in noisy images. In: Videometrics III, SPIE vol. 2350, Boston (1994) 136-147

[Pan et al., 1995] X. Pan, T.J. Ellis, T.A. Clarke, Robust tracking of circular features. Proc. BMVC95, Birmingham, UK, Sept. 1995, pp. 553-562.

[Panikolopoulos et al., 1992] N.P. Panikolopoulos, B. Nelson, P.K. Khosla, Monocular 3-D visual tracking of a moving target by an eye-in-hand robotic system. Proc. 31st Conference on Decision and Control, Tuscon Arizona, IEEE, December 1992, 3805-3810

[Porch, 1979] Porch, J., Gale, C., Utilisation of Closed Circuit Television in Sewers. Technical Report 109, Water Research Centre, Medmenham Laboratory, Medmenham, Bucks, SL7 2HD, UK. 1979

[Pretlove and Parker, 1993] Pretlove, J.R.G, Parker, G.A., An Interactive Industrial Robot Vision System. Int Jour of Patt Rec and Art Int, Vol 7, No.1, pp89-108, February 1993. (see also Pretlove, J.R.G., The Surrey Attentive Robot Vision System, In: Active Robot Vision: Camera Heads, Model-based Navigation and Reactive Control, Series in Machine Perception and Artificial Intelligence, Vol 6, Eds: Bowyer, K., Christensen, H., Bunke, H., Published by World Scientific Ltd, London, 1993.)

[Price, 1995] T. Price, Inspecting buried plastic pipe using a rotating sonic calliper. Proceedings of the 2nd International Conference on Advances in Underground Pipeline Engineering (1995) 126-137

- [Pridmore et al., 1987] Pridmore, T.P., J. Porrill and J.E.W. Mayhew, Segmentation and description of binocularly viewed contours. Proc. 2nd Alvey Vision Conference, Image and Vision Computing 5, No. 2, pp 132-138 (1987)
- [Ruiz-del-Solar and Köppen, 1996] J. Ruiz-del-Solar, M. Köppen, Sewage pipe image segmentation using a neural based architecture. Pattern Recognition Letters 17 (1996) 363-368
- [Sahoo et al., 1988] P.K. Sahoo, S. Soltani, A.K.C. Wong, Y.C. Chen, A survey of thresholding techniques. Computer Vision Graphics Image Process. 41 (1988), 233-260
- [Straforini et al., 1993] Straforini, M., C. Coelho and M. Campani, Extraction of vanishing points from images of indoor and outdoor scenes. Image and Vision Computing, 11, 2, (1993), 91-99.
- [Tai et al., 1992] Tai, A., J. Kittler, M. Petrou and T. Windeatt, Vanishing point detection. Proc. British Machine Vision Conference, (1992), 109-118.
- [Taylor et al., 1998] N. Taylor, T.P. Pridmore, S. Fu, Automatic visual detection of lateral junctions in sewers. Proc. Institn Civ. Engrs Wat., Marit. & Energy, 1988, 130, 56-69
- [TINA] Open Source Image Analysis Environment
<http://www.niac.man.ac.uk/Tina/>
- [Tsukiyama, 1995] Tsukiyama, T., Inferring the 3D shape formed by plane surfaces from isoluminance curves. Image and Vision Computing, 13:9, 671-681
- [Wallace, 1982] Wallace, A.M., Industrial applications of computer vision since 1982. IEE Proceedings, Vol. 135 part E, pp117-136, 1988.
- [Weil et al., 1994] G.J. Weil, R.J. Graf, L.M. Forister, Remote sensing pipeline rehabilitation methodologies based upon the utilisation of infrared thermography. Urban Drainage Rehabilitation Programmes and Techniques, ASCE (1994)

[Wirahadikusumah et al., 1998] R. Wirahadikusumah, D.M. Abraham, T. Iseley, R.K.Prasanth, Assessment technologies for sewer system rehabilitation. Automation in Construction 7 (1998) 259-270.

[WRC, 1986] Water Research Centre, Sewerage Rehabilitation Manual. Wrc Engineering Ltd., Swindon, UK.

[Xu et al., 1998] So K. Xu, A.R. Luxmore, T. Davies, Sewer pipe deformation assessment by image analysis of video surveys. Pattern Recognition, 31:2 (1998) 169-180

Appendix 2: Camera Head Kinematics and LINKLAB Simulation

Kinematic control for the camera head was modelled using LINKLAB, a simulation package that takes a simple description of the device to be controlled and generates the code required to control the system. LINKLAB requires a description of the system using Denavit-Hartenberg (D-H) labelling [Denavit and Hartenberg, 1955]. This is obtained as follows.

Figure A2.1(a) shows the camera and its mounting. Figure A2.1(b) gives a schematic representation of this system.

Figure A2.1(a). The camera and the mounting.

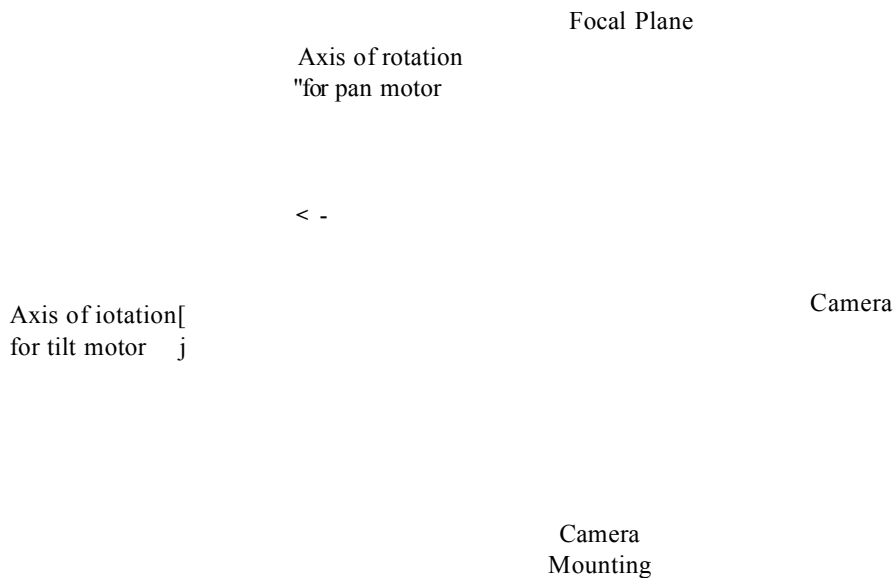


Figure A2.1(b). Schematic representation of the system in Figure A2.1(a). The end-effector is taken as being the in the focal plane of the camera.

The co-ordinate frame used to represent the system is given in Figure A2.2.

Figure A2.2. Co-ordinate frames used to represent camera and mount links and lengths in LINKLAB.

The individual co-ordinate systems ($i=0, \dots, 4$) are related one to the other. For revolute pairs, D-H notation relates the $(z-1)$ th co-ordinate frame to the z th co-ordinate frame by a transform obtained thus:

- i. rotate about axis z_{i-1} by θ_i (joint angle - the included angle of axes X_{i-1} and x_i)
- ii. translate along z_{i-1} by d_i (link offset - distance between the origin of the co-ordinate system $x_{i-1}, y_{i-1}, z_{i-1}$ and the foot of the common perpendicular)

- iii. translate along x_i by l_i (link length - distance between two feet of the common perpendicular)
- iv. rotate about x_i by α_i (link twist - the included angle of axes z_{i-1} and z_i)

(The transformation matrix is often denoted as $T(\theta_i, d_i, l_i, \alpha_i)$ for convenience.)

The transforms for the camera system are represented in Table A2.1. the distances d_p , d_b and d_t are those illustrated in Figures A2.1(b) and A2.2.

i	θ_i	d_i	l_i	α_i
1	0	0	0	$\pi/2$
2	$-\pi/2$	$-d_p$	0	$-\pi/2$
3	0	d_b	0	0
4	0	0	d_t	0

Table A2.1. LINKLAB scheme relating co-ordinate frames of Figure A2.2..

A full LINKLAB description of the system is given in the following file (Note: for the camera head in Figure A2.1(a), $d_p=83\text{mm}$, $d_b=81\text{mm}$ and $d_t=75\text{mm}$).

```
#####
#
# head_arm.rob
#
# Kinematic and dynamic parameter file. Kinematics are based on
# Denavit-Hartenberg labelling. Dynamics on the Newton-Euler
# equations. Robot structure is described as a pre-order
# recursive tree. For further details see 'Fundamentals of
# Robotics, Analysis and Control' (Schilling).
#
#####
# Version number
5
# Number of links
4
# Joint circle size
0.1
# Kinematic parameters per revolute link
# joint angle,
# joint distance,
# link length,
# joint twist angle,
# home position,
# max position,
# min position.

#The links are, in order :
#
0.0      0.0      0.0      1.5707  0.0      3.142   -3.142
-1.5707 -0.083    0.0      -1.5707  0.0      3.142   -3.142
0.0      0.081    0.0      0.0      0.0      3.142   -3.142
0.0      0.0      0.075    0.0      0.0      3.142   -3.142

# Recursive description of robot structure
1 1
2 1
3 1
4 0
```

```

# Dynamic parameters per link: link mass,
#   coefficient of viscous friction
#   coefficient of static friction
#   coefficient of dynamic friction
#   link centre of mass wrt link frame
#   link inertia tensor wrt link centre of mass

# Link 1
# mass, viscous, static, dynamic friction
1.0 0.0 0.0 0.0

# link centre of mass wrt link frame.
-0.5 0.0 0.0

# link inertia tensor wrt centre of mass of link. 3x3 matrix.
0.0 0.0 0.0
0.0 0.0833 0.0
0.0 0.0 0.0833

# Link 2
1.0 0.0 0.0 0.0

-0.5 0.0 0.0

0.0 0.0 0.0
0.0 0.0833 0.0
0.0 0.0 0.0833

# Link 3
1.0 0.0 0.0 0.0

-0.5 0.0 0.0

0.0 0.0 0.0
0.0 0.0833 0.0
0.0 0.0 0.0833

# Link 4
1.0 0.0 0.0 0.0

-0.5 0.0 0.0

0.0 0.0 0.0
0.0 0.0833 0.0
0.0 0.0 0.0833

#CSG representation of robot.
# parameters are:
#   node type {u=union, i=intersect, d=diff, n=null, c=cuboid, s=sphere}
#   x, y, z positions
#   yaw, pitch, roll angles
#   width, height, length dimensions

n
c 0 0.2 0 0 0 0 0.66 1.2 0.66
c 0.2 0 0 0 0 1.0 0.5 0.5
n

#
#####
EOF

```

LINKLAB allows simulation of the controller obtained from the system description given. Figures A2.3(a-b) serve to illustrate the camera simulation.

```

/03_axis_planar.ro**
Joint: 1 2 j3 4 5 6 7 c 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28
Position: -1^ -180 ■ ■ ■ 190
Set Hkm View Skeleton j CSC

/03_axis_planar.ro**
Joint: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 26
Position: 14, -180 -1. - - 190
Targets Fil* Snap Hk* Set lta view Skeleton j CSC

```

(a)

(b)

Figure A2.3. Illustration of LINKLAB simulation of camera manipulation.

Appendix Reference

[Denavit and Hartenberg, 1995] Denavit, J. and Hartenberg, R. S., "A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices," *ASME Journal of Applied Mechanisms*, 1955, pp. 215-221. (see also Yi Zhang, Introduction to Mechanisms, <http://www-2.cs.cmu.edu/People/rapidproto/mechanisms/chpt4.html>)

Appendix 3: TINA Code to Select Candidate Cracks.

The following code contains the filters (highlighted in red) which take the output of TINA's Canny edge detection routine (a list of linked lists) and select candidate cracks from the image. The unary filter progressively excludes single edges that do not meet the criteria described (see thesis). The pair-wise filter selects edge pairs that remain after the unary filter has been applied. For purposes of graphical illustration, the filters colour-code the edges as they are processed.

```
#include <tina/all_tina.h>
#include <tina/toolsfuncs.h>
#include <stdio.h>
#include <stdlib.h>

#include <tina/sys.h>
#include <tina/math.h>
#include <tina/vision.h>
#include <tina/tv.h>
#include <tina/tvfuncs.h>
#include <tina/tv_screen.h>
#include <xview/xview.h>
#include <xview/frame.h>
#include <xview/panel.h>
#include <xview/textsw.h>
#include <xview/font.h>
#include <xview/canvas.h>
#include <xview/cms.h>
#include <xview/xv_xrect.h>
#include <xview/openmenu.h>
#include <tina/tw.h>
#include <tina/Xvfuncs.h>
#include <tina/X11funcs.h>
#include <tina/draw.h>
#include <tina/drawfuncs.h>

static Tv *tv = NULL;
static Tv *graphtv = NULL;

static double estvpx = 128.0, estvpy = 128.0;
static int centx, centy, curvcount, use_gauss, all_strings;
static double *curv = NULL, *arcleng = NULL;
static double rhothresh = 50.0;
static double perpdist = 50;
static double pthresh = 50;
static int maxgap = 20;
static double disthresh = 50.0;
static double borderthresh = 50.0;
static double lengthhresh = 10.0;
static double tpercent = 80.0;
static Tw_callback *xbit = NULL;
static Tw_callback *ybit = NULL;
static Imrect *lines2_im = NULL, *er=NULL;
```

```

#define MAXPROF          1000 /* maximum gaussian profile size */

static int psize;          /* gaussian mask size */
static double sigma = 2.0; /* gaussian smoothing parameter */
static int border = 20;
static float profile[MAXPROF]; /* gaussian profile */
static float gprof(float, float);

static int in_fovea(Vec2 *pos)
{
    Imrect *image = mono_image_get();

    if (vec2_get_x(pos)<border || vec2_get_y(pos)<border ||
        image->region->ux-vec2_get_x(pos)<border ||
        image->region->uy-vec2_get_y(pos)<border)
        return(FALSE);
    else
        return(TRUE);
}

void string_display(Tv * tv, Tstring * string, int colour)
{
    Ddlist *start;
    Ddlist *end;
    Ddlist *dptr;
    Vec2 oldpos = {Vec2_id};
    Vec2 pos = {Vec2_id};

    if (string == NULL)
        return;

    start = string->start;
    end = string->end;

    tv_save_draw(tv);

    tv_color_set(tv, colour);

    oldpos = edge_image_pos((Edgel *) start->to);
    tv_point2(tv, oldpos);

    for (dptr = start; dptr != end;)
    {
        dptr = dptr->next;
        pos = edge_image_pos((Edgel *) dptr->to);
        tv_point2(tv, oldpos);
        oldpos = pos;
    }

    tv_reset_draw(tv);
}

double cube(double x)
{
    return(x*x*x);
}

static void plotcurv()
{
    int i, xcurr, ycurr, xnext, ynext, height, width, border;

```

```

double curvmax=0.0, arcmax=arcleng[curvcount-1];

/* first draw axes, leave a border of pixels */
height = tv_get_height(graphtv);
width = tv_get_width(graphtv);
border = 0.1*height;

tv_set_color(graphtv, white);
tv_linexy(graphtv, border, height-border,border,border);
tv_linexy(graphtv, border, height/2,width-border,height/2);
tv_textxy(graphtv, "arc length", width/2, height-(border/2));
tv_textxy(graphtv, "k", border/2, border/2);

/* now add curv[] */
for (i=0; i<curvcount; ++i)
    if (i==0 || fabs(curv[i])>curvmax)
        curvmax=fabs(curv[i]);

xcurr = border;
ycurr = height/2;
for (i=0; i<curvcount; ++i)
    {
        xnnext = border + (int)((arcleng[i]*(width-(2*border)))/arcmax);
        ynext = (height/2) -
            (int)((curv[i]*(height-(2*border))/2)/curvmax);
        tv_linexy(graphtv, xcurr, ycurr, xnnext, ynext);
        xcurr = xnnext;
        ycurr = ynext;
    }
}

static int curvthresh()
{
    double thold, max, absmax=0.0;
    int peakcount=0;
    int i, xleft, xright, y, *marks, height, width, border;

    marks = (int *)malloc(curvcount*sizeof(int));

    for (i=0; i<curvcount; ++i)          /* recompute maximum */
        if (i==0 || fabs(curv[i])>absmax)
            {
                absmax=fabs(curv[i]);
                max = curv[i];
            }

    thold = max*tpercent/100.0;

    if (!all_strings)                    /* only draw threshold line when dealing with */
        {                                /* selected strings */
            tv_set_color(graphtv, green);
            height = tv_get_height(graphtv);
            width = tv_get_width(graphtv);
            border = 0.1*height;

            xleft = border;
            xright = width-border;
            y = (height/2) - (int)((thold*(height-(2*border))/2)/absmax);

            tv_linexy(graphtv, xleft, y, xright, y);
        }
}

```

```

}

for (i=0; i<curvcount; ++i)          /* mark supra-thold values */
{
    if (max >= 0 && curv[i]>thold)
        marks[i] = 1;
    else if (max<0 && curv[i]<thold)
        marks[i] = 1;
    else
        marks[i] = 0;
}

for (i=0; i<curvcount; ++i)          /* count supra-thold values */
    if ((i==0 && marks[i]==1) ||
        (i>0 && marks[i-1]==0 && marks[i]==1))
        ++peakcount;

format("%d peaks detected\n", peakcount);

return(peakcount);
}

void string_curv(Tstring *string)      /* curvature as fn. of arc length */
{
    Ddlist *back, *curr, *front;
    Vec2 backpos = {Vec2_id};
    Vec2 currpos = {Vec2_id};
    Vec2 frontpos = {Vec2_id};
    Vec2 firstdiff = {Vec2_id};
    Vec2 secdiff = {Vec2_id};
    double backmag, frontmag, T, curvature, s=0.0;
    int i;

    if ((back=string->start)==NULL)
        error("string_curv() can't initialise back", non_fatal);
    if ((curr=back->next)==NULL)
        error("string_curv() can't initialise curr", non_fatal);
    if ((front=curr->next)==NULL)
        error("string_curv() can't initialise front", non_fatal);

    curv = (double *)malloc(string->count*sizeof(double));
    arcleng = (double *)malloc(string->count*sizeof(double));
    curvcount = 0;

    while (front!=string->end->next)
    {
        backpos = edge_image_pos((Edgel *) back->to);
        currpos = edge_image_pos((Edgel *) curr->to);
        frontpos = edge_image_pos((Edgel *) front->to);

        backmag = vec2_dist(backpos,currpos);
        frontmag = vec2_dist(currpos,frontpos);
        T = backmag/(backmag+frontmag);

        for (i=0;i<2;++i)
        {
            firstdiff.el[i] = backpos.el[i]*(T-1.0)/T +
                currpos.el[i]*(2.0*T-1.0)/(T*(T-1.0)) -
                frontpos.el[i]*T/(T-1.0);
            secdiff.el[i] = 2.0*(backpos.el[i]/T +

```

```

        currpos.el[i]/(T*(T-1.0)) -
        frontpos.el[i]/(T-1.0));
    }

    curv[curvcount] =
        vec2_cross(firstdiff,secdiff)/cube(vec2_mod(firstdiff));
    arcleng[curvcount] = (curvcount==0) ? backmag:
        arcleng[curvcount-1]+backmag;
    ++curvcount;

    back = back->next;
    curr = curr->next;
    front = front->next;
}
}

static void dogauss(Tstring *string)          /* Gaussian smoothing */
{
    int i,index;
    Vec2  backpos = {Vec2_id};
    Vec2  temppos = {Vec2_id};
    Vec2  frontpos = {Vec2_id};
    Vec2  frontlastpos = {Vec2_id};
    Vec2  backnextpos = {Vec2_id};
    Ddlist *back, *temp, *front;
    double sum[2],k=0.0,sampletot,cprof=profile[0],round;

    float *xs, *ys;
    int count = 0;

    xs = (float *)malloc(string->count*sizeof(float));
    ys = (float *)malloc(string->count*sizeof(float));

    for (temp=string->start;temp!=string->end->next;temp=temp->next)
    {
        temppos = edge_image_pos((Edgel *) temp->to);
        sum[0]=temppos.el[0]*cprof;
        sum[1]=temppos.el[1]*cprof;
        sampletot = cprof;

        back=temp->last;
        if (back!=NULL)
        {
            backpos = edge_image_pos((Edgel *) back->to);
            k = vec2_dist(temppos,backpos);
        }

        while (back!=NULL && k<(float)psize/10.0)
        {
            round = (k*10.0) - (int)(k*10.0);
            if (round>=0.5)
                index = (int)(k*10.0) + 1;
            else
                index = (int)(k*10.0);
            for (i=0;i<2;++i)
                sum[i] += profile[index]*(backpos.el[i]);
            sampletot += profile[index];

            back=back->last;

```



```

    if (back!=NULL)
        {
            backpos = edge_image_pos((Edgel *) back->to);
            backnextpos = edge_image_pos((Edgel *) back->next->to);
            k += vec2_dist(backnextpos,backpos);
        }
    }

    front=temp->next;
    if (front!=NULL)
        {
            frontpos = edge_image_pos((Edgel *) front->to);
            k = vec2_dist(temppos,frontpos);
        }
    while (front!=NULL && k<(float)psize/10.0)
        {
            round = (k*10.0) - (int)(k*10.0);
            if (round>=0.5)
                index = (int)(k*10.0) + 1;
            else
                index = (int)(k*10.0);
            for (i=0;i<2;++i)
                sum[i] += profile[index]*(frontpos.el[i]);
            sampletot += profile[index];

            front=front->next;
            if (front!=NULL)
                {
                    frontpos = edge_image_pos((Edgel *) front->to);
                    frontlastpos = edge_image_pos((Edgel *) front->last->to);

                    k += vec2_dist(frontlastpos,frontpos);
                }
        }

    xs[count]=sum[0]/sampletot; /* store new positions in arrays */
    ys[count]=sum[1]/sampletot;
    ++count;
}

for (temp=string->start, count=0;
     count<string->count && temp!=string->end->next;
     temp=temp->next, ++count)
    {
        Edgel *edge = (Edgel *)temp->to;

        edge->pos.el[0] = xs[count];
        edge->pos.el[1] = ys[count];
    }
}

static int getprof() /* put (half) a gaussian mask in profile[] */
{
    int i,psize;
    float sig2;

    sig2=2*sigma*sigma;

    for (i=0;i<MAXPROF;++i)

```

```

    {
        profile[i]=gprof((float)i,sig2);
        if (profile[i]<0.0001)
            break;
    }
    psize=i-1;

    return(psize);
}

static float gprof(float x, float s2)
{
    float i,sum=0;
    short tot=0;

    x *= 0.1;
    for (i=x-0.05;i<=x+0.05;i+=0.01)
    {
        sum+=exp(-i*/s2);
        tot++;
    }

    return(sum/tot);
}

static void markpoint(int x, int y)
{
    Imrect *image;
    float drawx, drawy;
    int xcoord, ycoord, height, width, imwidth, imheight;

    if ((image = mono_image_get())==NULL)
        error("markpoint: null image", non_fatal);

    height = tv_get_height(tv);
    width = tv_get_width(tv);
    if (height!=width)
        error("non-square tv in markpoint", warning);

    imwidth = image->region->ux - image->region->lx;
    imheight = image->region->uy - image->region->ly;

    xcoord = x - imwidth/2;
    ycoord = -(y - imheight/2);

    /* scale image to window width */
    drawx = ((float)width/(float)imwidth)*x;
    drawy = ((float)width/(float)imwidth)*y;

    if (imheight != imwidth) /* centre vertically */
        drawy = drawy + (height-imheight*((float)width/(float)imwidth))/2.0;

    tv_set_color(tv, white);
    tv_cross(tv, ipos(drawx, drawy), 20);
    tv_circle(tv, ipos(drawx, drawy),
              (int)(perpdist*(float)width/(float)imwidth));
}

static double getrho(Ddlist *start, Ddlist *end)

```

```

{
float theta, or, rho, x, y, vx, vy;
Vec2  startpos = {Vec2_id};
Vec2  endpos = {Vec2_id};
Imrect *im = mono_image();

startpos = edge_image_pos((EdgeI *) start->to);
endpos = edge_image_pos((EdgeI *) end->to);

centx = (im->region->ux - im->region->lx)/2;
centy = (im->region->uy - im->region->ly)/2;
markpoint(centx, centy);

x = vec2_x(startpos) - centx; /* shift origin to VP estimate */
y = centy - vec2_y(startpos); /* rho calculation assumes y +ve upwards */

vx = vec2_x(endpos) - vec2_x(startpos);
vy = vec2_y(endpos) - vec2_y(startpos); /* vector orig in top left corner 7

if (vx<0)
{
    vx = -vx;
    vy = -vy;
}
or = atan(fabs((double)vy/(double)vx));
or = (vy >= 0) ? PI - or : or;

theta = or + PI/2.0;
if (theta > PI)
    theta = theta - PI;

rho = x*cos(theta) + y*sin(theta);

return(rho);
}

```

/* Unary Test Filter7

```

int  string_filter(Tv * tv, Tstring * string)
{
Ddlist 'start;
Ddlist 'end;
Ddlist 'dptr;
Vec2  oldpos = {Vec2_id};
Vec2  pos = {Vec2_id};
int peaks, near=0, maxlength = (int)(sqrt(2.0) * (double)border);
double percent;

if (string == NULL)
    return (FALSE);

start = string->start; /* decide whether string is in right area */
end = string->end;

oldpos = edge_image_pos((EdgeI *) start->to);
if (in_fovea(&oldpos))
    ++near; /* count points within region of interest */

for (dptr = start; dptr != end;)
{

```

```

    dptr = dptr->next;
    pos = edge_image_pos((EdgeI *) dptr->to);
    if (inJovea(&pos))
        ++near;
    oldpos = pos;
}

if (string->count - near < borderthresh)
{
    if (tv != NULL)
        string_display(tv, string, blue);
    return(FALSE); /* not in periphery */
}
else if (string->count < lengthresh || string->count > maxlength)
{
    if (tv != NULL)
        string_display(tv, string, green);
    return(FALSE); /* in periphery but too short */
}
else if (fabs(getrho(start, end)) > perpdist)
{
    if (tv != NULL)
        string_display(tv, string, );
    return(FALSE); /* straight line between end points 7
                    /* does not pass close to image centre 7
}
else
{
    if (tv != NULL)
        string_display(tv, string, red); /* one of possible pair7
    return(TRUE);
}
}

void string_list_filter(Tv *tv, List *stringJist)
{
    /* derived from tv_stringJist 7
    List *sptr;
    Tstring 'copy;

    if (tv == NULL || tv->tv_screen == NULL || stringjist == NULL)
        return;

    for (sptr = stringjist; sptr != NULL; sptr = sptr->next)
    {
        /* apply following to each string 7
        copy = str_copy(sptr->to, (void *)(*edge_copy), NULL);
        if (use_gauss)
            dogauss(copy);
        stringJilter(tv, copy);
    }

static void filter_proc(void)
{
    Imrect *er = mono_edges();
    List 'stringjist;

    if (all.strings)
        stringjist = (List *) prop_get(er->props, STRING);
    else
        stringjist = seg_select_es_get();
}

```

```

    if (use_gauss)
        psize = getprof();
    tv_save_draw(tv);
    string_list_filter(tv, stringjist);
}

/*edge pair filter*/
void pairJilter(Tv *tv, Tstring * string!, Tstring * string2)
{
    Ddlist *dptr1, *dptr2;
    Vec2 pos1 = {Vec2Jd};
    Vec2 pos2 = {Vec2Jd};
    Vec2 pos3 = {Vec2Jd};
    Vec2 pos4 = {Vec2Jd};
    int near = 0;
    double dist, mindist, percent;
    Imrect *image = mono_image_get();
    float drawx, drawy;
    int height, width, imwidth, imheight;
    float meanx, meany;

    if (string1 == NULL || string2 == NULL)
        return;

    for (dptr1 = string1->start; dptr1 = dptr1->next; dptr1 != NULL)
    {
        pos1 = edge_image_pos((Edgel *) dptr1->to);
        mindist = -1.0;

        for (dptr2 = string2->start; dptr2 = dptr2->next; dptr2 != NULL)
        {
            pos2 = edgeJmage_pos((Edgel *) dptr2->to);
            if (mindist<0.0 || (dist = vec2__dist(pos1, pos2)) < mindist)
                mindist = dist;
        }

        if (mindist < maxgap)
            ++near;
    }

    percent = ((double)near/((double)(string1->count))*100.0;

    if (percent > pthresh)
    {
        string_display(tv, string1, cyan); /*edge pair is candidate crack*/
        string_display(tv, string2, cyan);

/* compute and mark the centroid of the 2 edges*/
        pos1 = edgeJmage_pos((Edgel *) string1->start->to);
        pos2 = edge_image_pos((Edgel *) string1->end->to);
        pos3 = edge_image_pos((Edgel *) string2->start->to);
        pos4 = edge_image_pos((Edgel *) string2->end->to);

        meanx = (vec2_get_x(&pos1) + vec2_get_x(&pos2) + vec2_get_x(&pos3) +
vec2_get_x(&pos4))/4.0;
        meany = (vec2_get_y(&pos1) + vec2_get_y(&pos2) + vec2_get_y(&pos3) +
vec2_get_y(&pos4))/4.0;

```

```

    height = tv_get_height(tv);
    width = tv_get_width(tv);
    if (height!=width)
        error("non-square tv in pair_filter", warning);

    imwidth = image->region->ux - image->region->lx;
    imheight = image->region->uy - image->region->ly;

    /* ===== */

    /* scale image to window width 7
    drawx = ((float)width/(float)imwidth)*meanx;
    drawy = ((float)width/(float)imwidth)*meany;

    if (imheight != imwidth) /* centre vertically 7
        drawy = drawy + (height-imheight*((float)width/(float)imwidth))/2.0;

    tv_set_color(tv,cyan);
    tv_cross(tv, ipos(drawx, drawy), 20); /*mark cross hairsV
    tv_circle(tv, ipos(drawx, drawy),20);
}
}
void pairJistJilter(Tv *tv, List *stringjist)
{
    List *sptr1, *sptr2;

    if (tv == NULL || tv->tv_screen == NULL || stringjist == NULL)
        return;

    stringjistJilter(tv, stringjist); /* apply unary tests 7

    for (sptr1 = stringjist; sptr1 != NULL; sptr1 = sptr1->next)
        if (stringJilter(NULL, sptr1->to))
            for (sptr2 = sptr1->next; sptr2 != NULL; sptr2 = sptr2->next)
                { /* apply following to each string 7
                    if (stringJilter(NULL, sptr2->to))
                        pairjilter(tv, sptr1->to, sptr2->to);
                }
}

static void pairs_proc(void)
{
    Imrect *er = mono_edges();
    List 'stringjist;

    if (allstrings)
        stringjist = (List *) prop_get(er->props, STRING);
    else
        stringjist = seg_select_es_get();

    pairJistJilter(tv, stringjist);
}

static void showestVP_proc(int choice)
{
    Imrect 'image;

```

```

float drawx, drawy;
int type, height, width, imwidth, imheight;

if (choice==1 || choice==3)
{
    if ((image = mono_image_get())==NULL)
        error("showestVP_proc: null image", non_fatal);
}
else if (choice==2)
{
    if ((Bool) stack_check_types(IMRECT, NULL) == true)
        image = (Imrect *)stack_pop(&type);
    else
        error("showestVP_proc: no image on stack", non_fatal);
}
else
    error("showestVP_proc: unknown choice", non_fatal);

if (choice==1 || choice==2)
    tv_imrect2(tv, image);    /* show image on tv */

height = tv_get_height(tv);
width = tv_get_width(tv);
if (height!=width)
    error("non-square tv in showestVP", warning);

imwidth = image->region->ux - image->region->lx;
imheight = image->region->uy - image->region->ly;

/* scale image to window width */
drawx = ((float)width/(float)imwidth)*estvpx;
drawy = ((float)width/(float)imwidth)*estvpy;

if (imheight != imwidth) /* centre vertically */
    drawy = drawy + (height-imheight*((float)width/(float)imwidth))/2.0;

tv_set_color(tv, blue);
tv_cross(tv, ipos(drawx, drawy), 20);
tv_circle(tv, ipos(drawx, drawy),
           (int)(rhothresh*(float)width/(float)imwidth));
}

static int isolated(Imrect *im, int x, int y)
{
    int i, j, lx, ux, ly, uy;

    lx = im->region->lx;
    ux = im->region->ux;
    ly = im->region->ly;
    uy = im->region->uy;

    for (i=x-1; i<=x+1; ++i)
        for (j=y-1; j<=y+1; ++j)
            if ((i!=x || j!=y) &&
                i>=lx && i<ux && j>=ly && j<uy &&
                im_get_pixf(im, j, i)==0)
                return(0);    /* pixel is not isolated */

    return(1);    /* pixel is isolated */
}

```

```

static void relabel(lmrect *im, float old, float new)/* replace old with new */
{
    int x, y, lx, ux, ly, uy;

    lx = im->region->lx;
    ux = im->region->ux;
    ly = im->region->ly;
    uy = im->region->uy;

    if (old==0.0)
        error("relabel asked to relabel background", non_fatal);

    for (x=lx; x< ux; ++x)        /* done over the whole im, inefficient */
        for (y=ly; y< uy; ++y)    /* but easy */
            if (im_get_pixf(im, y, x)==old)
                im_put_pixf(new, im, y, x);
}

static int neigh_labels(lmrect *im,int x, int y, float *label) /* return */
{
    /* no. of distinct neighbouring labels, pass back one via label */
    int lx, ux, ly, uy, distinct, count=0, row, i, j;
    float min=0.0, val[4], record[4];

    /* only concerned with previous 4 pixels on raster scan */
    /* put them into a local array for ease */

    lx = im->region->lx;
    ux = im->region->ux;
    ly = im->region->ly;
    uy = im->region->uy;

    val[0] = (y-1<ly) ? 0 : im_get_pixf(im, y-1, x);
    for (row=y-1, i=1; row<=y+1; ++row, ++i)
        val[i] = (x-1<lx || row>=uy || row<ly) ? 0.0 : im_get_pixf(im, row, x-1);

    for (i=0; i<4; ++i)        /* count and record distinct labels */
        if (val[i]!=0.0)
            {
                distinct = 1;
                for (j=0; j<i; ++j)
                    if (val[j] == val[i])
                        distinct = 0;

                if (distinct==1)
                    {
                        record[count] = val[i];
                        ++count;
                    }
            }

    if (count == 0)
        return(count);
    else if (count == 1)
        {
            if (record[0]==0.0)
                error("neigh_labels returning zero B", non_fatal);

            *label = record[0];
            return(count);
        }
}

```



```

    }
else /* return lowest distinct label */
{
    for (i=0; i<count; ++i)
        if (record[i]>0.0 && (min==0.0 || record[i]<min))
            min = record[i];

    if (min==0.0)
        error("neigh_labels returning zero A", non_fatal);

    *label = min;
    return(count);
}
}

static float big_region(Imrect *im, int count)
{
    int x, y, lx, ux, ly, uy, i, j, k, found;
    Imrect *copy;
    float *sizes, *labels, label, maxlabel, max=0.0;

    if (im==NULL)
        error("big_region : null image passed", non_fatal);
    if (count==0)
        error("big_region : zero count", non_fatal);

    copy = im_copy(im);

    labels = (float *)malloc(count*sizeof(float));
    sizes = (float *)malloc(count*sizeof(float));

    lx = im->region->lx;
    ux = im->region->ux;
    ly = im->region->ly;
    uy = im->region->uy;

    for (i=0; i< count; ++i) /* destructive count of pixels with ith label */
    {
        found = 0; /* first find the first pixel */
        for (x=lx; x<ux && found==0; ++x)
            for (y=ly; y<uy && found==0; ++y)
                if ((label=im_get_pixf(copy, y, x)) != 0.0)
                {
                    found = 1;
                    labels[i] = label;
                    sizes[i] = 1; /* first pixel found */
                }

        /* now resume scan and cont pixels marked label */
        for (j=lx; j<ux; ++j)
            for (k=ly; k<uy; ++k)
                if (im_get_pixf(copy, k, j)==labels[i])
                {
                    sizes[i] += 1; /* first pixel found */
                    im_put_pixf(0.0, copy, k, j); /* and destroyed */
                }
    }

    /* now find and return label of largest region */
    for (i=0; i< count; ++i)

```

```

    if (i==0 || sizes[i]>max)
    {
        max = sizes[i];
        maxlabel = labels[i];
    }

    format("region %f is largest\n", maxlabel);
    return(maxlabel);
}

static void estVP_proc(void) /* assumes binary image on the stack, black */
{
    /* pixels/regions are interesting */
    Imrect *im_orig, *regions;
    int context, row, numregions=0;
    int x, y, lx, ux, ly, uy, type, current=0, count, spots=0;
    float xsum=0.0, ysum=0.0, label, val, largest, dist, maxdist=0.0;

    if ((Bool) stack_check_types(IMRECT, NULL) == true)
    {
        im_orig = (Imrect *) stack_pop(&type);

        lx = im_orig->region->lx;
        ux = im_orig->region->ux;
        ly = im_orig->region->ly;
        uy = im_orig->region->uy;

        for (x=lx; x< ux; ++x) /* first remove isolated black pixels */
        for (y=ly; y< uy; ++y)
            if (im_get_pixf(im_orig, y, x)==0 && isolated(im_orig,x,y))
            {
                ++spots;
                im_put_pixf(255.0, im_orig, y, x);
            }
        format("%d isolated pixels removed\n", spots);

        /* now segment the image into regions */
        regions = im_alloc(im_orig->height, im_orig->width, im_orig->region,
                           im_orig->vtype);

        for (x=lx; x< ux; ++x) /* clear region map */
        for (y=ly; y< uy; ++y)
            im_put_pixf(0.0, regions, y, x);

        for (x=lx; x< ux; ++x) /* label regions */
        for (y=ly; y< uy; ++y)
            if (im_get_pixf(im_orig, y, x)==0)
            {
                /* determine region number of black pixels */
                context = neigh_labels(regions, x, y, &label);

                if (context == 0) /* no neighbouring labelled pixels */
                {
                    /* so new region */
                    ++current;
                    ++numregions;
                    im_put_pixf((float)current, regions, y, x);
                }
                else if (context == 1) /* 1 neighbouring label, extend it */
                    im_put_pixf(label, regions, y, x);
                else
                {
                    /* > 1 label; regions merging, label is lowest label */

                    if (y-1 >= ly )

```

```

        {
            val = im_get_pixf(regions, y-1, x);
            if (val!=0.0 && val!=label)
                {
                    relabel(regions, val, label);
                    --numregions;
                }
        }

    for (row=y-1;row<=y+1;++row)
        if (x-1>=lx && row<uy && row>=ly)
            {
                val = im_get_pixf(regions, row, x-1);
                if (val!=0.0 && val!=label)
                    {
                        relabel(regions, val, label);
                        --numregions;
                    }
            }
        }
    }

format("%d regions found\n", numregions);

/* find largest region */
largest = big_region(regions, numregions);

count = 0;
for (x=lx; x< ux; ++x)          /* and compute centroid */
for (y=ly; y< uy; ++y)
    if (im_get_pixf(regions, y, x) == largest)
        {
            xsum += (x+0.5); /* assume intensity measured at */
            ysum += (y+0.5); /* pixel centre */
            ++count;
        }
    else /* black out other regions */
        im_put_pixf(0.0, regions, y, x);

estvpx = xsum/count;
estvpy = ysum/count;
tw_fglobal_reset(xbit);
tw_fglobal_reset(ybit);

/* also set filter threshold at distance of furthest region member */
/* estimated VP */
for (x=lx; x< ux; ++x)
for (y=ly; y< uy; ++y)
    if (im_get_pixf(regions, y, x) == largest)
        {
            dist = sqrt(((float)x+0.5-estvpx)*((float)x+0.5-estvpx) +
                ((float)y+0.5-estvpy)*((float)y+0.5-estvpy));
            if (dist > maxdist)
                maxdist = dist;
        }

rhothresh = maxdist;

regions = imf_scale(regions, 0.0, 255.0);
stack_push(regions, IMRECT, NULL);

```

```

    }
    else
    {
        error("estVP_proc : wrong type on stack", non_fatal);
    }
}

static void mono_init(Tv *tv)
{
    Imrect *im = monoJimage();

    if (im != NULL)
        tv_camera2_image(tv, im->width, im->height);
}

static void tv_choice_proc(int choice)
{
    switch (choice)
    {
        case 1:
            tv_set_init(tv, monojnit);
            tv_set_next(tv);
            break;
        case 2:
            tv_set_init(graphtv, monojnit);
            tv_set_next(graphtv);
            break;

        default:
            error("tv_choice_proc : unknown choice\n", warning);
            break;
    }
}

static void tv_display_proc(int choice)
{
    Imrect *image = NULL;

    switch (choice)
    {
        case 1:
            image = imf_scale(lines2Jm, 0.0, 255.0);
            tv_imrect2(tv, image);
            break;
        default:
            error("tv_display_proc : unknown choice\n", warning);
            break;
    }
}

static void use_gauss_proc(int val)

    use_gauss = val;

static void string_choose_proc(int val)

    allstrings = val;

```

```

void *crack_tool(int xpos, int ypos)
{
    static void 'tool;

    tv = tv_create("Lateral");
    graphtv = tv_create("Lateral Graphics");

    tool = (void *) tw_tool("Crack Tool", xpos, ypos);
    {
        tw_menubar("Setup",
            "Install TVs",
                "Lateral", tv_choice_proc, 1,
                "Lateral Graphics", tv_choice_proc, 2,
            NULL,
        NULL);

        tw_menubar(NULL,
            "Display",
                "Lateral", tv_display_proc, 1,
            NULL,
        NULL);

        tw_newrow();
        tw_button("Estimate VP", estVP_proc, NULL);
        tw_menubar(NULL,
            "Show estimate",
                "on image", showestVP_proc, 1,
                "on region", showestVP_proc, 2,
                "on grey", showestVP_proc, 3,
            NULL,
        NULL);

        tw_newrow();
        xbit = tw_fglobal("estimated VP x    &estvpx, 12);
        ybit = tw_fglobal("estimated VP y    &estvpy, 12);

        /* tw_newrow();
        tw_choice("Gaussian Smoothing:", use_gauss_proc, 1,
            "Off", "On", NULL);*/

        tw_newrow();
        tw_choice("Strings:", string_choose_proc, 1,
            "Selected", "All", NULL);

        tw_newrow();
        tw_button("Unary Filter", filter_proc, NULL);
        tw_button("Pairwise Filter", pairs_proc, NULL);

        /* tw_newrow();
        tw_fglobal("sigma :", &sigma, 12);

        tw_newrow();
        tw_fglobal("curvature thold factor (%):", &tpercent, 12);*/

        tw_newrow();
        tw_iglobal("border:", &border, 12);

        tw_newrow();
        tw_fglobal("length in border :", &borderthresh, 12);

        tw_newrow();

```

```
        tw_fglobal("length threshold    &lengthresh, 12);  
        tw_newrow();  
        tw_jglobalf'rho threshold    &perpdist, 12);  
    }  
tw_end_tool();  
return tool;
```