Machine Vision and Applications (2004) 16: 47–58
Digital Object Identifier (DOI) 10.1007/s00138-004-0156-3

Machine Vision and Applications

CeDAR: A real-world vision system

Mechanism, control and visual processing

Andrew Dankers, Alexander Zelinsky

Robotic Systems Laboratory, Research School of Information Sciences and Engineering, The Australian National University, Canberra ACT 0200 Australia

Published online: 25 October 2004 –

Abstract. We report on the development of a multi-purpose active visual sensor system for real-world application. The Cable-Drive Active-Vision Robot, *CeDAR*, has been designed for use on a diverse range of platforms to perform a diverse range of tasks. The novel, biologically inspired design has evolved from a systems-based approach. The mechanism is compact and lightweight and is capable of motions that exceed human visual performance and earlier mechanical designs. The control system complements the mechanical design to implement the basic visual behaviours of fixation, smooth pursuit and saccade, with stability during high-speed motions, high precision and repeatability. Real-time algorithms have been developed that process stereo colour images, resulting in a suite of basic visual competencies. We have developed a scheme to fuse the results of the visual algorithms into robust task-oriented behaviours by adopting a statistical framework. CeDAR has been successfully used for experiments in autonomous vehicle guidance, object tracking and visual sensing for mobile robot experiments.

Keywords: Active vision – Vision processing – Multiple-cue – Particle filter

1 Introduction

In recent years, increased hardware performance versus component cost has brought vision firmly into the realm of practical robot sensors. The domain of computer vision has sufficiently matured to enable researchers to build and experiment with systems that model and interact with that which they observe. We concern ourselves with refining a multi-purpose visual sensor system for real-world, real-time, task-directed robotic applications. The vision system must be fit for use on a diverse range of platforms , performing a diverse range of tasks. It must be able to intelligently gather data from its environment in a sufficiently timely fashion for it to make the decisions for task-oriented behaviour.

Correspondence to: A. Zelinsky (e-mail: alex@syseng.anu.edu.au)

A practical system is required to react to the real world in real time. However, real environments contain events occurring at all timescales. It is therefore practical to consider real time as a time period commensurate with the defined task. The real world is an unstructured, possibly cluttered, dynamic environment that extends beyond sensor range. A vision system operating in the real world must therefore be equipped with mechanisms to fixate its attention upon that which is important within the time fr ame of its relevance while simultaneously disregarding background irrelevancies. Attention should be directed to what can loosely be described as *interesting*, including that which may be useful to the completion of goal-directed behaviour. The sensory system must therefore be capable of shifting focus to the location of interest and maintaining focus even if the target is moving. In addition, as the environment is large, it may be necessary for the architecture of the vision system to allow it to ext end its field of awareness via searching behaviour. Appropriate sensory and processing resources must be selected with these considerations in mind.

Such requirements point strongly to the use of a visual system able to adjust its visual parameters to aid task-oriented behaviour, an approach labelled *active* [3] or *animate* [5] vision. An active vision approach can offer impressive computational benefits for scene analysis in realistic environments [4]. We believe that active vision is one of the best sensing modalities for task-oriented interaction with the real world.

Here we begin by introducing the CeDAR mechanism and its control philosophies. We then present the framework we have adopted for visual processing. Then, examples of machine vision applications that utilise these philosophies are presented, including quantitative results.

2 CeDAR's evolution

A multi-modal systems approach was adopted where the mechanism (Fig. 1) and its control and vision processing modules were developed in parallel, with integration in mind [6]. From the outset it was desired that the system be capable of out-performing existing synthetic vision systems and replicating the abilities of the human vision system while incorporating reasonably sized payloads (e.g. two 700-g cameras). The mechanism and control modules have been conceived with the

Fig. 1. CeDAR (Cable-Drive Active-Vision Robot) mechanism

purpose of having a high level of mechanical performance to allow rapid motion and short reaction times, as well as being re-configurable for application to many situations with minimal modification. The control and vision processing modules have been developed for use with standard digital visual hardware operating at a 30-Hz frame rate. Low cost, ease of reproduction and configurability for mobility were also significant factors in the evolution of the design.

3 Mechanism

The concept of controlled camera movements to facilitate computer vision is a regime analogous to various vision systems observed in biology, and with few exceptions, animals have developed active visual abilities. The human eye achieves extraordinary performance through its low-weight and low-inertia muscle actuation. Accordingly, existing synthetic vision systems have been built that endeavour to mimic the properties of biological vision systems.

3.1 Related work

A brief overview of recent active vision devices reveals a trend towards smaller, more agile systems. In the past the goals were to experiment with different configurations using large systems with many degrees of freedom like the KTH active head [20] with its 13 degress of freedom and Yorick 11-14 [21] with a 55-cm baseline[2] and re-configurable joints. Although useful for experimentation, these systems were excessively cumbersome for agile motion and not easily configurable for mobility. More recently, developers of smaller active heads such as the palm-sized Yorick 5-5C [21] and ESCHeR [16] with an 18-cm baseline have reported on lightweight systems suitable for mobile robot navigation and for teleprescence applications. Internationally, the trend towards smaller active vision systems comparable to the size of the human head is pushing the limit of motor, gear box and camera design. In many systems, the size of the motors and cameras limits the compactness of the active head and the motors themselves add to the inertia of the moving components. A notable exception is the Agile Eye [11], where no motor carries the mass of any other. All three

[2] Distance between camera retinas.

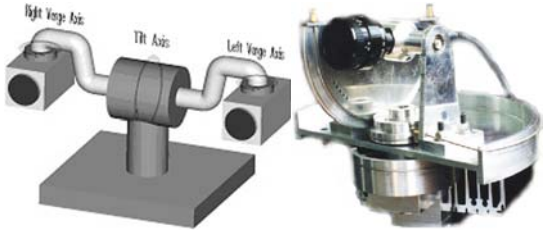

Fig. 2. *Left*: The Helmholtz configuration. *Right*: Early prototype

versions of Yorick as well as ESCHeR use harmonic or gear drive technology. A limitation of the technology is an unavoidably large speed-reduction ratio that limits the output speed to less than 100 rpm. We aim to improve upon such past mechanical designs to manifest a mechanism that is lightweight and more agile than previous approaches.

3.2 Mechanical design

The mechanism has incorporated aspects of design from other laboratories and from the mechanics of the human visual system. As seen in nature, muscles are lightweight, exhibit high accelerations and do not suffer significantly from the common problem encountered in serial active head designs whereby each degree of freedom requires sufficiently powerful actuators to move all previous degrees of freedom, including their actuators. It was shown that by relocating the motors to a fixed base and thereby reducing the inertia of the active component to essentially a camera, problems inherent in serial design were alleviated [7]. This biologically inspired concept is an important feature of CeDAR's novel mechanical design.

Additionally, because backlash-free speed reduction is essential for high-speed performance, the choice of transmission system for the parallel architecture is important. During high-speed movements such as saccades[3] – where motors are driven at maximum acceleration – velocity saturation for harmonic-drive gear boxes is of concern. Cable drive is a novel alternative for use with repeated bounded motions that does not induce speed limitations, operates lubricant-free with low friction, exhibits high torque transmission and is low cost.

An earlier prototype [7] (Fig. 2) proved the usefulness of cable drive transmissions and parallel mechanical architectures in a two-degree-of-freedom active "eye" system. The prototype was fast (able to achieve an angular velocity of $600°\mathrm{s}^{-1}$ for each axis), responsive (angular accelerations of up to $72,000°\mathrm{s}^{-2}$) and accurate (to a resolution within $0.01°$). In 2000, the prototype's architecture was transferred to a stereo Helmholtz configuration [19] (Fig. 2), resulting in the present mechanical design of CeDAR (Fig. 3). Actuation has been transferred through cable drive circuits that seamlessly integrate with the parallel architecture. Power ratings for the actuators (70 W tilt axis, 20 W each verge axis) are such that the unit can perform normally in mobile robotic situations, where power consumption is of concern.

An important kinematic property of the design is that the axes intersect at the optical centre of each camera, minimising kinematic translational effects. This property of the stereo

[3] Saccade is the ability to rapidly transfer fixation from one visual target to another.

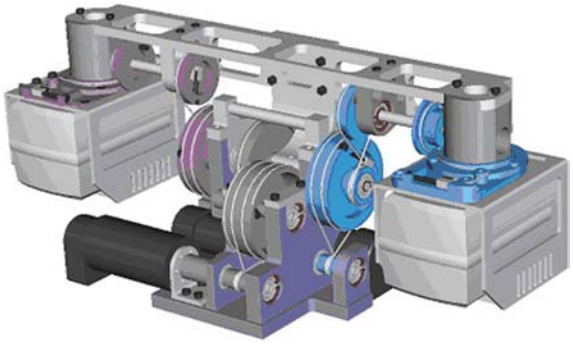

Fig. 3. CAD model rear view of CeDAR showing parallel architecture and cable drive

camera configuration aids development of stereo algorithms such as depth re-construction through image disparity calculations, a competency that some biological vision systems exhibit [27].

3.3 Mechanical performance evaluation

The performance figures traditionally reported for active vision mechanisms consist of maximum angular velocity, maximum angular acceleration, angular resolution and axis range. While the latter two are highly relevant, we consider the others to be not especially useful in that they do not detail any form of specific task competency. Additional specifications for an active vision system that not only involve the speed and acceleration of the axes but also express the usage intention of the system in the form of a functional requirement are given in Table 1.

The tabled values need to be achieved in order to satisfy constraints related to desired motion abilities. CeDAR's maximum allowable full-speed saccade time was required to be $0.18\,\mathrm{s}$ to enable three 90° gaze-shift saccades, with an allowance for each to be preceded by four target location video frames and succeeded by one stabilisation video frame per second. Just over 5 frames are captured during the saccade itself.

The minimum allowable full-speed stop-to-stop angular change within one video frame was required to be 15°, which equates to the ability to track an object moving past the cameras at up to $4\,\mathrm{ms}^{-1}$ at a distance of $1\,\mathrm{m}$. The angular resolution has been selected with the aim of allowing the platform to perform meaningfully small camera movements and to allow single pixel selection.

The required maximum range, payload and baseline specifications were based on the desire to use commonly available motorised-zoom cameras; however, the unit is easly re-configured to incorporate many off-the-shelf cameras. Of course, as there exists the potential to incorporate smaller cameras, improvements in performance are likely. The saccade rate and pointing accuracy were chosen based on the desired performance of the device in its intended applications. Real-time tracking was the most basic desired task.

Speed performance was determined by driving the joints to their maximum range, speed and acceleration in a cyclical fashion (repeated saccades). The command positions and actual positions of the joints were logged at millisecond intervals. The position data were then differentiated using a three-point rule and filtered using a seven-point moving average to obtain velocity and acceleration profiles.

A series of accuracy tests were also conducted using laser pointers mounted on the head. Repeatability, the ability to return to an absolute position after a series of complex movements, was demonstrated by moving the joints to an arbitrary position, relocating to another location and then returning to the original point. In systems that suffer from backlash, friction or poor compliance, the return point differs from the original. Angular resolution, the smallest angle that can be actuated, was measured by moving the joints a minimal increment. Coordinated motion, the joints' ability to move in unison in both time and space, was demonstrated by verging both laser pointers to the same location on a wall, then commanding the system to follow a predetermined trajectory. Coordination was evaluated according to how closely the lasers were converged throughout the motion.

Table 1 lists results of the accuracy tests along with results from the speed tests and the design specifications. All of the mechanical design specifications were met. CeDAR's mechanical performance compares favourably to similar systems (Table 2).

4 Control

Gaze control can be broken down into several basic tasks: gaze fixation, saccade and smooth pursuit[3] [19]. CeDAR's

Table 1. Performance specifications and test results

	Test		Specification	
Specification	Tilt	Vergence	Tilt	Vergence
Max. velocity	$600^\circ\mathrm{s}^{-1}$	$800^\circ\mathrm{s}^{-1}$	$600^\circ\mathrm{s}^{-1}$	$600^\circ\mathrm{s}^{-1}$
Max. acceleration	$18,000^\circ\mathrm{s}^{-2}$	$20,000^\circ\mathrm{s}^{-2}$	$10,000^\circ\mathrm{s}^{-2}$	$10,000^\circ\mathrm{s}^{-2}$
Saccade rate	$5\mathrm{s}^{-1}$	$6\mathrm{s}^{-1}$	$5\mathrm{s}^{-1}$	$5\mathrm{s}^{-1}$
Ang. repeatability	0.01°	0.01°	0.01°	0.01°
Ang. resolution	0.01°	0.01°	0.01°	0.01°
Max. range	90°	90°	90°	90°
Payload	Two 700-g cameras			
Baseline	30 cm			

[3] Smooth pursuit involves gaze fixation upon a moving target

Table 2. Performance of world-class vision systems (mass includes payload)

	Max. Vel.	Max. Accel.	Approx. Mass
CeDAR	$800°s^{-1}$	$20{,}000°s^{-2}$	3.5 kg
Yorick 8-11	$600°s^{-1}$	$38{,}000°s^{-2}$	9.0 kg
Yorick 85CR	$660°s^{-1}$	$10{,}000°s^{-2}$	3.0 kg
ESCHeR	$400°s^{-1}$	$16{,}000°s^{-2}$	2.0 kg
Agile eye	$1,000°s^{-1}$	$20{,}000°s^{-2}$	
KTH	$180°s^{-1}$		15 kg

control routines are an extension of work undertaken by [19] on trapezoidal profile motion (TPM). In particular, our approach allows for the implementation of a single algorithm for both saccade and smooth pursuit, enhancing the simplicity and compactness of the controller design.

The essence of the TPM problem is to detect and transfer attention to the desired target gaze point some distance from the image centre either in the shortest time possible (saccade) or as smoothly as possible (smooth pursuit). Both the joints' and target's starting velocity are potentially non-zero and disparate. Specifically, we cause each visual axis to accelerate constantly to a calculated ceiling velocity[4], coast at this velocity for a given period, then decelerate at the same constant rate as the acceleration until the target velocity is reached (Fig. 4). Mathematically, it is a 4D problem per axis where the acceleration a, ceiling velocity v, move time T and total distance traveled x are unknown. The initial joint velocity v_1, target velocity v_2 and the target's initial distance from the image centre x_0 are the givens.

If the acceleration a is assumed to be constant, the time taken by the head to accelerate from its initial velocity to the ceiling velocity is

$$T_a = \frac{sv - v_1}{sa}, \tag{1}$$

where s is positive for $v_1 < v$ and negative for $v_1 > v$. Similarly, the time to decelerate to target velocity is

$$T_d = \frac{sv - v_2}{sa}. \tag{2}$$

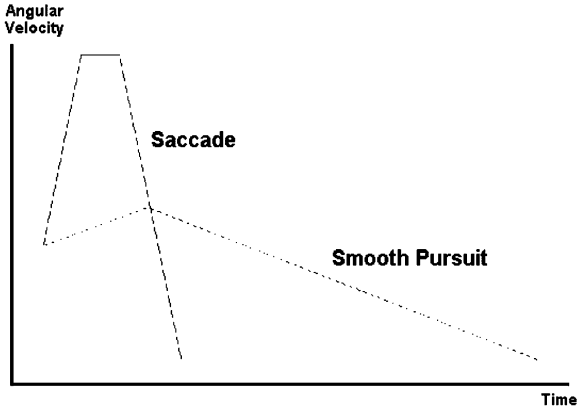

Fig. 4. Trapezoidal profile motion velocity profiles

[4] The maximum absolute velocity of the TPM trajectory

Note that acceleration and deceleration rates are equal. If T_c is the time spent coasting at the ceiling velocity, the total time for TPM is

$$T = T_a + T_c + T_d. \tag{3}$$

The distance traveled by the head in time T is

$$x = \frac{sv + v_1}{2}T_a + svT_c + \frac{sv + v_2}{2}T_d, \tag{4}$$

but it can also be considered as the sum of the initial distance of the target from the foveal centre x_0 and the distance traveled by the target during the move

$$x = x_0 + Tv_2. \tag{5}$$

These general equations can be used to develop the case for saccade and smooth pursuit.

Saccade involves changing the head's current position and velocity state to that of the target, as inferred by its previous states, in the shortest time possible. Motion smoothness is not a concern and hence acceleration is set to its maximum possible magnitude. Two cases can arise:

- The ceiling velocity required for the action is less than the maximum allowed velocity, and hence no time is spent coasting.
- The theoretical ceiling velocity required for the action is greater than the maximum allowed velocity, and hence some time must be spent coasting.

It is useful to assume T_c is initially zero so that T can be deduced from Eqs. 1–5 with sv calculated as

$$sv = v_2 \pm \frac{1}{2}\sqrt{4sx_0a - 2(v_1^2 + v_2^2 - 4v_1v_2}, \tag{6}$$

where the smaller value is taken for $v_2 > v_1$ and vice versa. If v exceeds the maximum allowed velocity, a and v are replaced by their maxima. Then

$$T_c = \frac{kT_a - x_0}{v_2 - sv}, \tag{7}$$

where

$$k = sv - \frac{v_1 + v_2}{2} \tag{8}$$

is calculated to deduce T. Equation 6 also defines the value of s so that the operand of the radical is greater than or equal to zero:

$$s = \begin{cases} 1 \text{ for } (v_1 - v_2)^2 + 4x_0a \geq 0\,, \\ -1 \text{ otherwise.} \end{cases} \tag{9}$$

Smooth pursuit involves moving from one position and velocity state to the next in a given amount of time with optimal smoothness. To achieve this, the acceleration in moving to and from the ceiling velocity must be as small as possible. Again both the coasting and non-coasting cases are relevant. With the assumption that the coasting velocity is initially zero, Eqs. 1–5 yield

$$v = \frac{x}{T} \pm \frac{1}{2T}\sqrt{4x^2 - 4Tx(v_1 + v_2) + 2T^2(v_1^2 + v_2^2)}. \tag{10}$$

If these values are in excess of the maximum allowable velocity of the head, the time constraint is unrealisable. In this instance, a saccade is initiated.

5 Visual processing

Our previous work [6] revealed that a comprehensive collection of basic visual behaviours may be more suitable for investigation of an unknown, complex, unstructured environment – such as the real world – than a high-level processing technique constructed for a specific situation. An objective is the implementation of such a suite of elementary competencies. As low-level mechanisms, the algorithms should attempt to minimise assumed knowledge in the form of explicit world or object models and high-level decision making. However, they should be designed in such a way as to be bases suitable for having these abstractions applied on top of them. They should be general purpose in order to maximise their applicability to real-world layouts and situations and be suitable for use either in serial or parallel. Finally, the algorithms should be designed for use in combination while exhibiting real-time performance by addressing the trade-offs between accuracy and speed.

5.1 Cue extraction

Competencies that have been implemented include motion segmentation, edge detection, depth mapping (implementation as per [13]), zero-disparity filtering [8], colour detection (implementation as per [17]), radial symmetry (implementation as per [17]) and template matching. Additionally, permutations of these basic cues, and others, enhance the suite of visual competencies.

With the aid of such competencies, relevant information can be extracted from a scene so that gaze fixation can be effected. As an example of this most basic information extraction, the system has successfully used single cues to actively track the template of facial features and/or coloured objects [18].

The use of multiple cues in parallel has enabled more robust object tracking. A multiple-cue object-tracking algorithm has been implemented that incorporates four simple cues –

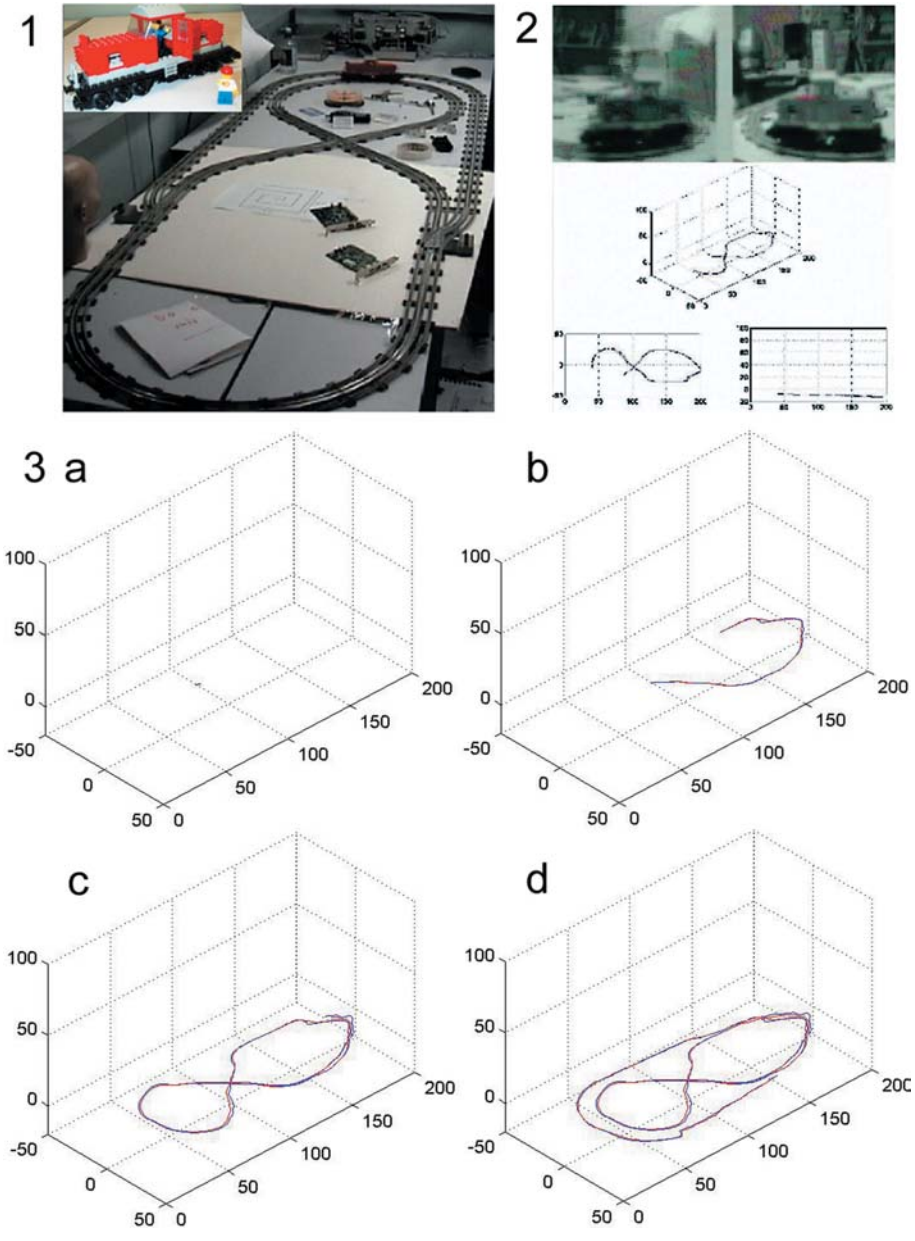

Fig. 5. (1) Scenario. (2) Screenshot: left and right camera views and trajectory. (3) Target trajectory at $t = 0\,\text{s}$, $10\,\text{s}$, $20\,\text{s}$ and $30\,\text{s}$ (units in cm)

colour, edge detection, texture detection and motion. A cue voting scheme was adopted to identify pixel locations in the view frame that appeared target-like with respect to each cue. A simple zero-disparity filter using virtual horopters[5] then visually e xtracted the object from its surroundings, as well as mapping its position in 3D space. The processing of all visual information took, on average, only 8 ms per frame on a dual Pentium III computer – well below the 30-Hz frame rate set by the digital camera image capture frequency. Unfiltered operation was susceptible to distractions due to target-like regions in the camera's views. Kalman filtering reduced the effect of these distractions significantly. The algorithm allows successful real-time tracking of arbitrary objects through a cluttered environment (Fig. 5).

5.2 Robust cue fusion

Intuitively, however, it is not sensible to assume all cues or competencies should be relied upon equally for different tasks in changeable environments. And it is clear that no single cue can perform reliably in all situations. An efficient and robust vision system will intelligently combine information from a number of different cues whilst effectively managing the available computational resources. In deciding which cues should be used and when, how the cues should be combined, and how much computational resources should be expended, the regime must:

- Efficiently allocate finite computational resources when calculating cues, accounting for the cues' expected utility and resource requirements.
- Facilitate cues running at different frequencies.
- Locate a target in multi-dimensional state space, e.g. determine the target's 3D location and orientation.
- Allow tracking of multiple hypotheses.

5.3 Related work

A number of researchers have utilised multiple cues to detect and track, for example, people in scenes; however, there have been few attempts (nonetheless, [Spengler] is an example of one group who address this issue) to develop a system that considers the allocation of finite computational resources amongst the available cues.

Recent work by Soto and Khosla [22] presents a system based on *intelligent agents* that adaptively combines multi-dimensional information sources (*agents*) to estimate the state of a target. A particle filter is used to track the target's state, and metrics are used to quantify the performance of the agents. Initial results for person tracking in 2D show the potential of this particle-filter-based approach.

Triesh and von der Malsburg [25] present a system suitable for combining a number of cues to track a target's 2D location in an image. The output of each sensor is compared to a prototype describing the target (a face) with respect to that sensor. An adaptive weighting is given to each cue based on the cue's performance over recent frames, and the final result for each frame is determined as the weighted sum of the probability images. The system adapts to targets with changing appearance by dynamically updating the prototypes based on the sensor outputs at the target location over recent frames.

5.4 The adaptive cue fusion architecture

Consequently, a framework has been developed to dynamically allocate computational resources over multiple cues. Bayesian theory (Sect. 5.5) provides the framework for cue fusion, and resource scheduling is used to intelligently allocate the limited computational resources across the suite of cues (Sect. 5.7.2). Each cue's expected utility and resource requirement is taken into account, and the system can accommodate for cues running at different frequencies to allow cues perceived to be performing less constructively to be run slowly in the background. A particle filter [12] (Sect. 5.6) is adopted to maintain multiple hypotheses of the target location and orientation in 3D state space.

Ultimately, a number of cues are calculated from image and state information and combined to provide evidence strengthening or attenuating the belief in each hypothesis. Figure 6 shows the structure of the system. It consists of two subsystems: a particle filter and a cue processor, each of which cycles through its loops once per frame. These subsystems interact as shown by the thick arrows: the particle filter passes the current particle locations to the cue processor, and the cue processor determines probabilities for the particles and passes these back to the particle filter.

5.5 Bayesian approach

Given a state space of possible target poses, the problem of target localisation can be expressed probabilistically as the estimation of the posterior probability density function over the space of possible poses based on the available data. That is, at time t estimate the posterior probability $P(s_t|e_{0\ldots t})$ of a state s_t given all available evidence $e_{0\ldots t}$ from time 0 to t.

Using Bayesian probability theory and applying the *Markov assumption*[6] the desired probability $P(s_t|e_{0\ldots t})$ can be expressed recursively in terms of the current evidence and knowledge of the previous states. This is referred to as *Markov localisation* and is represented mathematically by the following equation:

$$P(s_t|e_{0\ldots t}) = \eta_t P(e_t|s_t) \sum_{s_{t-1}} P(s_t|s_{t-1}) P(s_{t-1}|e_{0\ldots t-1}), \quad (11)$$

where η_t is a constant normaliser that ensures the probabilities sum to one, $\eta_t = 1/P(e_{0\ldots t-1})$.

The derivation, as detailed by Thrun [24], sequentially applies Bayes rule, the Markov assumption, the theorem of total probability and the Markov assumption again, and is as follows:

[5] The horopter defines the locus of points in space that map to a stereo image pair with identical coordinates.

[6] The *Markov assumption* states that future events are independent of the past, given that the current state is known.

$$
\begin{aligned}
P(s_t|e_{0\ldots t}) &= \eta_t P(e_t|e_{0..t-1}, s_t) P(s_t|e_{0\ldots t-1}) \\
&= \eta_t P(e_t|s_t) P(s_t|e_{0\ldots t-1}) \\
&= \eta_t(e_t|s_t) \\
&\quad \sum_{s_{t-1}} P(s_t|e_{0\ldots t-1}, s_{t-1}) P(s_{t-1}|e_{0\ldots t-1}) \\
&= \eta_t P(e_t|s_t) \\
&\quad \sum_{s_{t-1}} P(s_t|s_{t-1}) P(s_{t-1}|e_{0\ldots t-1}).
\end{aligned}
$$

This formulation allows fusion of the information provided by any of the adopted cues.

5.6 Particle filter

The particle filter approach to target localisation, including the condensation algorithm [12] and Monte Carlo localisation [24], uses a large number of particles to *explore* the state space. Each particle represents a hypothesised target location in state space. Initially the particles are uniformly randomly distributed across the state space, and for each subsequent frame the algorithm cycles through the steps illustrated in Fig. 6:

1. Deterministic drift: Particles are moved according to a deterministic motion model.
2. Probability density function (PDF) update: The probability for every new particle location is determined.
3. Particle resampling: 90% of the particles are resampled with replacement such that the probability of choosing a particular sample is equal to the PDF at that point; the remaining 10% of particle are distributed randomly throughout the state space.
4. Particle diffusion: Particles are moved a small distance in state space under Brownian motion.

This results in particles accumulating in regions of high probability and dispersing from other regions; thus the particle density indicates the most likely target states. See [12] for a comprehensive discussion of this method.

The primary appeals of the particle filter approach to localisation and tracking are its scalability (computational requirement varies linearly with the number of particles) and its ability to deal with multiple hypotheses and thus more readily recover from tracking errors. However, the particle filter was applied here for several additional reasons:

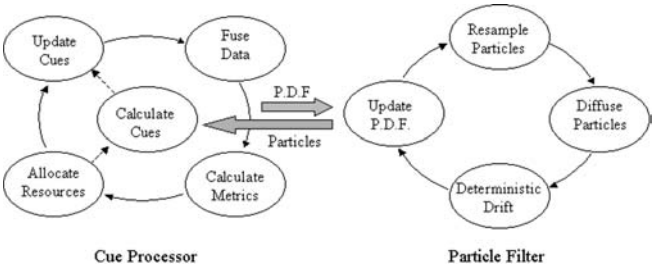

Fig. 6. Adaptive fusion architecture

- It provides an efficient means of searching for a target in a multi-dimensional state space.
- It reduces the search problem to a verification problem, i.e. is a given hypothesis face-like according to the sensor information?
- It allows fusion of cues running at different frequencies.

The last point is especially important for a system operating multiple cues with limited computational resources as it facilitates running some cues slower than frame rate (with minimal computational expense) and incorporating the result from these cues when they become available.

If a cue takes n frames to return a result, by the time the cue is ready, the particles will have moved from where they were n frames ago. To facilitate such cues, the system keeps a record of every particle's history over a specified number of frames (k). The cue value determined for a particle $n \leq k$ frames ago can then be assigned to the children of that particle in the current frame, thus propagating forward the cue's response to the current frame. Conversely, probabilities associated with particles that were not propagated are discarded.

5.7 Cue processor

The cue processor deals with the calculation and fusion of cues. It also determines metrics measuring the performance of each cue and the allocation of computational resources to individual cues. Two metrics, the Kullback–Leibler divergence and the uncertainty deviation [22], are used to evaluate the performance of each cue with respect to the fused result and the individual result of the cue respectively.

For each frame the cue processor cycles through the steps illustrated in Fig. 6:

1. Update cues: Access recently calculated cues.
2. Fuse data: Fuse the results of different cues to estimate the overall probability for each hypothesised target state.
3. Calculate metrics: Determine the metrics quantifying the performance of each cue on the last image frame.
4. Allocate resources: Based on the anticipated performance of the individual cues, allocate computational resources to maximise the quality of information obtained.

The *calculate cues* component of the system accepts requests for cue measurements and handles the requests using only the quantity of computational resource allocated to it by the *allocate resources* component. Some calculations may take longer than a single frame, but, as discussed in the previous section, the *update PDF* component is able to accommodate these slow cues and propagate their effect through to the current probability values.

5.7.1 Quantifying cue performance

The performance, or *utility*, of each active cue is estimated every frame and used to decide the distribution of computational resources across the cues (Sect. 5.7.2).

Fusing the results of all available cues is assumed to give the best estimate of the true PDF $P(e_t|s_t)$ across the state space. So the performance of the jth cue can be quantified by measuring how closely the cue's PDF $P(e_{j,t}|s_t)$ matches

$P(e_t|s_t)$. This can be done using the relative entropy, or the Kullback–Leibler distance [15], an information theoretic measure of how accurate an approximation one PDF is to another, given by

$$\delta_t\left(P(e_t|s_t), P(e_{j,t}|s_t)\right) = \sum_{s_t} P(e_t|s_t) \log \frac{P(e_t|s_t)}{P(e_{j,t}|s_t)},$$

where s_t is the particle states at time t. Soto and Khosla [22] used this metric to rate the performance of their cues, and Triesh and von der Malsburg [25] considered it, but opted for a simpler ad hoc measure.

The utility of the jth cue at time t is defined as

$$u_t(j) = -\delta_t(P(e_t|s_t), P(e_{j,t}|s_t)).$$

5.7.2 Resource allocation

The computational resources of the system are dynamically allocated based on the performance metrics that predict the future performance of each cue. This configuration not only optimises the performance of the cues for the current situation, as it dynamically chooses the most suited cues to the current conditions, but it also makes the system flexible to future changes in hardware and software.

The operation of the resource allocator is a simple process of searching through the complete space of possible cue combinations for the one that has the best overall utility. The overall utility of a combination of cues is the sum of the performance metrics of each cue.

A certain fraction of the time between each frame is devoted to cues running at frame rate, while the rest of the time is devoted to those cues that run at speeds less than frame rate. The performance metric of a cue running at a rate slower than frame rate is reduced exponentially by a discount factor for each frame it is late. The discount factor was introduced on the premise that a result obtained over 8 frames is worth less than one that is obtained over 2 frames.

The resource allocator starts by generating all cue combinations that can run in the time allocated for cues running at frame rate. It then chooses the combination with the best overall utility. A list of all combinations of the remaining cues over all possible slower frame rates is generated such that no combination exceeds the time allocated for the slower cues. Initially the slower rates were set to once every 2, 4, and 8 frames. Taking into account the discount factor for slow cues, the combination that has the best overall utility is chosen.

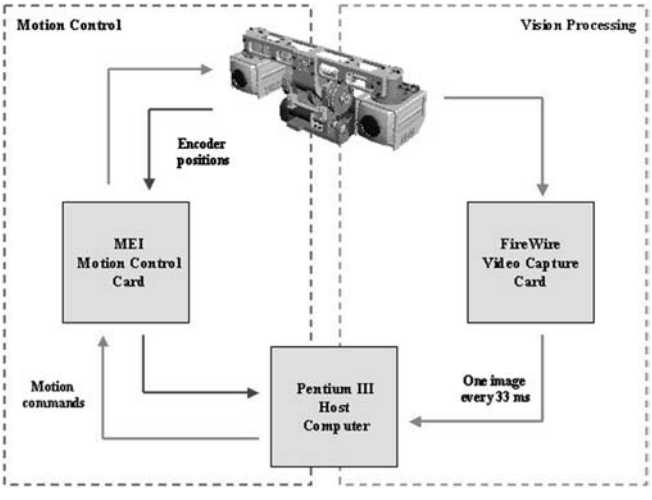

Fig. 7. System integration

5.8 Vision processing operation

The system has been implemented with two uncalibrated colour stereo video cameras as sensors. The images from these cameras undergo some preprocessing and are then passed to the cues, where each target location hypothesis is tested by computation of all active cues.

Preprocessing is only performed once for each new set of images, whereas hypothesis testing requires one test for every target hypothesis generated by the particle filter. The preprocessing required for each frame is governed by the cues that are to be computed.

Each particle from the particle filter presents a hypothesis target location in state space. Using a pinhole camera model and the generic head model in Fig. 9a, the size, location and orientation of each hypothesis are determined in the image. All active cues are calculated for each hypothesis.

Each cue returns a set of probabilities $P(e_j|s_t)$ indicating the ith active cue's belief in the jth hypothesis. These probabilities are fused to determine the overall belief in the jth hypothesis b_j as follows:

$$P(e|s_t) = \prod_j \left(P(e_j|s_t)(1-\alpha) + \alpha\right),$$

where $\alpha \in (0,1)$ is included to prevent a zero value for a single $P(e_j|s_t)$, forcing $P(e|s_t)$ to zero. In this paper $\alpha = 0.1$ was used.

6 System integration

The interaction between the motion control and vision processing routines is summarised in Fig. 7. A firewire card captures images from each of the Sony DFW-VL500 cameras every 33 ms. The images are processed to determine the desired gaze. The gaze controller routines calculate the desired visual trajectory and visual parameters. An MEI (Motion Engineering Inc.) or Servo To Go (Servo To Go Inc.) card delivers actuation instructions to CeDAR via a PWM (pulse width modulation) amplifier and acquires positional feedback.

7 Task-specific applications

We now elaborate quantitatively upon two instances where the vision processing schemes described here have been implemented: face tracking (Sect. 7.1) and lane tracking (Sect. 7.2). Figure 8 shows CeDAR used in an auonomous vehicle project and as a sensor for a mobile robot.

Fig. 8. Examples of task-specific applications. *Left*: Autonomous vehicle project. *Right*: Sensing for mobile robot applications

7.1 Face tracking

Face tracking involves localising a face using video frames of a scene. Here, the system is used to track the location and pose of a person's face in 3D space as it moves in a cluttered environment.

7.1.1 Relevant cues

The cues were chosen on the grounds of simplicity and efficiency. All cues use the head model dimensions shown in Fig. 9a. In the following descriptions the *face region* and *face boundary* refer respectively to the light and dark grey regions in Fig. 9b.

1. **Elliptical skin region cue** returns the average skin likelihood of the pixels within the face region.
2. **Skin detector cue** returns 0.5 if any of the pixels sampled in the face region had skin likelihood values within the top 10% of values in the current skin likelihood image, and 0 otherwise.
3. **Non-skin boundary cue** returns a high value if there are few skin colour pixels in the face boundary region.
4. **Radially symmetric skin cue** is the value of appropriate (as determined by the hypothesised depth) skin-based radial symmetry image at the target location.
5. **Radially symmetric intensity cue** is the value of appropriate (as determined by the hypothesised depth) facial symmetry image at the target location.
6. **Radially symmetric eye cue** is the value of appropriate (as determined by the hypothesised depth) radial symmetry image at the hypothesised eye locations.
7. **Eye location cue** uses integral projection [14] to search the regions in Fig. 9c of the intensity image for the darkest bands aligned with the horizontal axis of the head. A high value is returned if these are close to the hypothesised eye locations.
8. **Head depth cue** compares the depths in the face region with the hypothesised depth of the target, returning a high value when these are in agreement.
9. **Head boundary depth cue** compares the depths in the face boundary region to the hypothesised target depth, giving a high value when these are different.

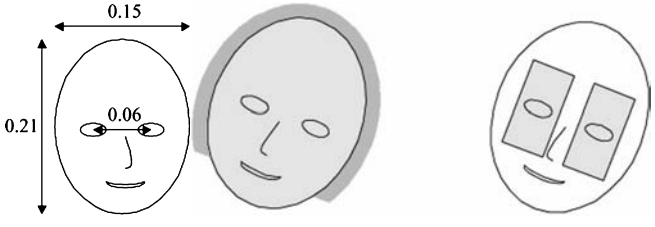

Fig. 9. a Generic head target with dimensions in meters. **b** Elliptical face region (*light*) and face boundary region (*dark*). **c** Search regions for integral projection

7.1.2 Implementation

The performance of the system was demonstrated tracking a human face in a cluttered scene. A sample frame of the sequence is presented in Fig. 10 along with particle distributions. Cues were dynamically scheduled to run once every 1, 2, 4, or 8 frames according to their calculated utility and computational cost.

The simplicity of the cues means no one cue is able to reliably track the head in 3D space; however, by fusing multiple cues the ambiguity in the target location is reduced. Furthermore, by adaptively rescheduling the cues the system is able to enhance the tracking performance possible under a given resource constraint.

7.2 Lane tracking

Lane tracking involves developing an understanding of the position of a driven vehicle with respect to the road along

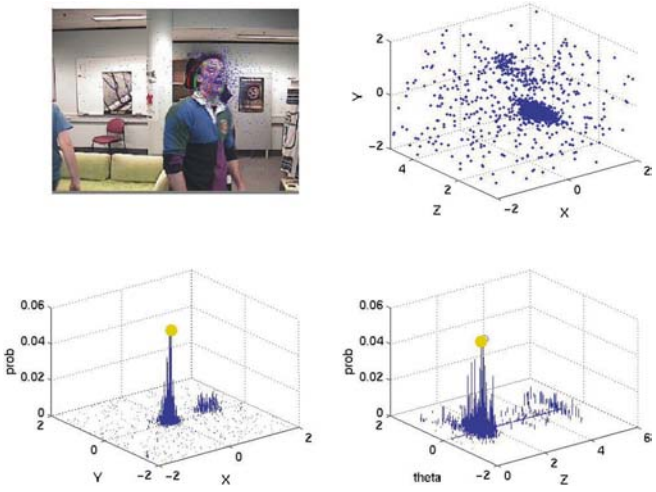

Fig. 10. Frame in a face tracking sequence showing (clockwise) particles in image and in 3D space and particle distributions over x, y and z, θ states

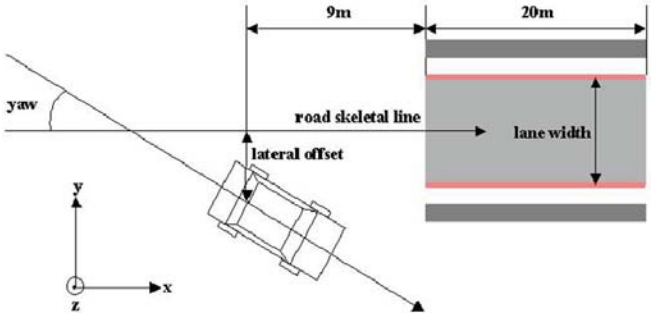

Fig. 11. Road model used for the first phase particle filter. The *dark shaded region* is used as the non-road boundary in the colour cues, while the *light shaded region* is the road region. Note that the figure is exaggerated for clarity

which it travels using video frames of a scene. Despite many impressive results from lane trackers in the past [2,9,23,26], it is clear that no single cue can perform reliably in all situations. Here we adopt the prescribed methodology to track the lateral offset and the yaw of a vehicle relative to the skeletal line of the road.

A dual-phase particle filter system was used to reduce the search space for the lane tracker. The first particle filter searches for the road width, the lateral offset of the vehicle from the centreline of the road and the yaw of the vehicle with respect to the centreline of the road. The second particle filter captures the horizontal and vertical road curvature in the mid- to far-field ranges using the state information captured by the first particle filter. In the work reported in this paper, the second phase particle filter of the system was not used, and no road curvature was calculated.

The state space for the particle filter is the lateral offset of the vehicle relative to the skeletal line of the road, the yaw of the vehicle with respect to the skeletal line and the road width (Fig. 11).

7.2.1 Relevant cues

Each cue is specifically developed to work independently from the other cues and is customised to perform well under different situations (i.e. edge-based lane marker tracking, colour-based road tracking, etc.).

The cues chosen for this experiment were designed to be simple and efficient while being suited to a different set of road scenarios. Individually, each of the cues would perform poorly, but when combined through the cue fusion process they produce a robust solution to lane tracking. Each cue listed below uses the road model shown in Fig. 11 to process the probability of each hypothesis from the particle filter.

1. **Lane marker cue** is designed for roads that have lane markings. A modified ternary correlation[7] to preprocess an intensity image of the road, and the cue returns the average value of the pixels along the hypothesised road edges.
2. **Road edge cue** is suited to roads with lane markings or roads with defined edges. It uses a preprocessed edge map and returns the average value of the pixels along the hypothesised road edges.
3. **Road colour cue** is useful for any roads that have a different colour than their surroundings (both unmarked and marked roads). It returns the average pixel value in the hypothesised road region from a colour probability map that is dynamically generated each iteration using the estimated road parameters from the previous iteration.
4. **Non-road colour boundary cue** is the opposite of the road colour cue and returns the average road colour probability of the non-road regions.
5. **Road width cue** is particularly useful on multi-lane roads where it is possible for the other cues to see two or more lanes as one. It returns a value from a Gaussian function centred at a desired road width given the hypothesised road width. The desired road width used in this cue was 3.61 m, which was empirically determined from previous lane tracking experiments to be the average road width.
6. **Elastic lane cue** is used to move particles towards the lane that the vehicle is in. It returns 1 if the lateral offset of the vehicle is less than half of the road width and 0.5 otherwise.

[7] The 1D ternary correlation function is modified to be two-sided with a step from -1 to 1 and back to -1.

7.2.2 Implementation

Figure 12 shows particle distribution plots of the perceived road width and yaw in a frame from the lane tracker in operation.

The lane tracker was tested in several different scenarios including:

- Highway driving with light traffic,
- Outer-city driving with high curvature roads,
- Inner-city driving with moderate levels of traffic.

Figure 13 shows the output of the lane tracker in the above scenarios using the six different cues.

The lane tracker was found to work robustly and solved the problems typically associated with lane tracking including:

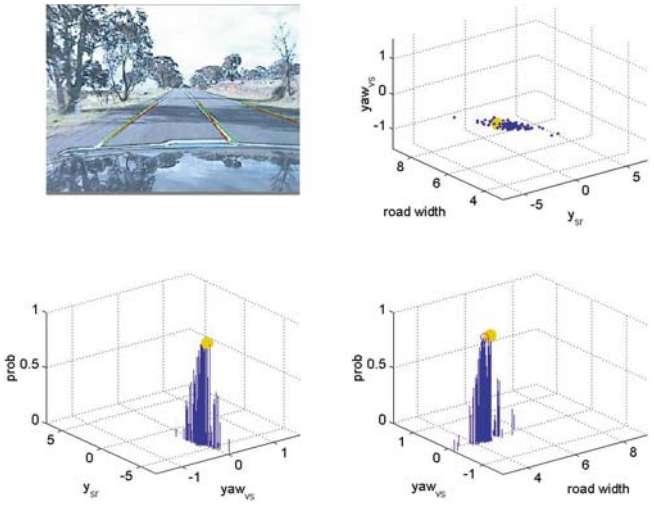

Fig. 12. Frame in a lane tracking sequence showing (clockwise) the perceived lane, lane width and orientation, yaw and offset, and yaw and width distributions

Fig. 13. Results from the lane tracker. The *boxes* indicate the end of the lines that mark the tracked lane

- Dramatic lighting changes (panel a in Fig. 13).
- Changes in road colour (panels d–f in Fig. 13).
- Shadows across the road (panels c, g–h in Fig. 13).
- Roads with miscellaneous lines that are not lane markings (panel i in Fig. 13).
- Lane markings that disappear and reappear.

This can be attributed to the combination of particle filtering and cue fusion. Because of the particle filter, cues only have to validate a hypothesis and do not have to search for the road. This indirectly incorporates a number of a priori constraints into the system (such as road edges meeting at the vanishing point in image space and the edges lying in the road plane) which assist it in its detection task.

Cue fusion was found to dramatically increase robustness due to the variety of conditions the cues were suited to. The final configuration of cues in the system is a direct result of earlier experiments uncovering certain conditions in which the cues would fail.

8 Future work

Already three CeDAR active heads are in operation at the Robotic Systems Laboratory.

A framework that permits the operation of any passive camera algorithms on an active-vision platform has been implemented and is undergoing refinement. Active depth and optical flow cues, for example, become more dense and accurate than their passive camera counterparts. The use of this framework for scene analysis, mobile robot guidance and obstacle and collision avoidance constitutes future work.

The development of additional visual cues for integration with the adaptive cue fusion architecture will also continue. Salency and conspicuity maps, as well as various other visual cues and combinations of cues, are expected to extend the existing set of visual cues, coinciding with the implementation of new task-oriented behaviors.

Furthermore, the adaptive cue fusion architecture is expected to be extended. Future work will focus on spawning Kalman filters around well-condensed targets, allowing the particles to be redistributed across the state space, like a drift net, to search for and track other potential targets.

9 Conclusion

Philosophies important to the evolution of a multi-purpose active visual sensor system for real-world application have been presented. Our multi-modal approach has resulted in a high-performance visual agent that integrates high-performance mechanical, control and vision processing architectures.

We have developed a biologically inspired, novel mechanical design that compares favourably with state-of-the art active mechanisms. Generalised trapezoidal profile motion control routines effect stable, precise and repeatable gaze control. A suite of basic visual competencies has been created, and a statistical framework for allocating computational resources across multiple cues for general task-specific behaviours has been developed. Guidelines for task competency performance evaluation of the system have been presented. The vision system has been successfully used in object tracking, lane tracking for autonomous vehicle guidance, and visual sensing for mobile robots.[3]

Web footage

Footage from work with the CeDAR system can be found on the RSL demonstrations page:

`http://www.syseng.anu.edu.au/rsl`

Acknowledgements. The authors would like to acknowledge the significant contribution made by Gareth Loy and Nicholas Apostaloff for their assistance with implementation and experimentation.

References

1. Truong H, Abdallah S, Rougeaux S, Zelinsky A (2000) A novel mechanism for stereo active vision. In: Proc. Australian conference on robotics and automation

2. Batavia PH, Pomerleau DA, Thorpe CE (1997) Overtaking vehicle detection using implicit optical flow. In: Proc. IEEE conference on transport systems
3. Aloimonos J, Weiss I, Bandopadhay A (2988) Active vision. Int J Comput Vis 1:333–356
4. Bajczy R (1988) Active perception. Proc IEEE 76(8):996–1005
5. Ballard D (1991) Animate vision. Artif Intell 48:57–86
6. Brooks A, Abdallah S, Zelinsky A, Kieffer J (1998) A multimodal approach to real-time active vision. In: International conference on intelligent robotics
7. Brooks A, Dickins G, Zelinsky A, Kieffer J, Abdallah S (1997) A high-performance camera platform for real-time active vision. In: 1st international conference on field and service robotics
8. Coombs D, Brown C (1992) Real-time smooth pursuit tracking for a moving binocular robot. In: Proc. international conference on computer vision and pattern recognition, pp 23–28
9. Dickmanns ED (1999) An exception-based, multi-focal, saccadic (ems) vision system for vehicle guidance. In: Proc. international symposium on robotics and research
10. Fletcher L, Apostoloff N, Chen J, Zelinsky A (2001) Computer vision for vehicle monitoring and control. In: Australian conference on robotics and automation
11. Gosselin C, St.-Pierre E, Gagne M (1996) On the development of the agile eye. IEEE Robot Automat Mag 29–37
12. Isard M, Blake A (1998) Condensation – conditional density propagation for visual tracking. Int J Comput Vis 29(1):5–28
13. Kagami S, Okada K, Inaba M, Inoue H (2000) Design and implementation of onbody real-time depthmap generation system. In: IEEE conference on robotics and automation
14. Kanade T (1973) Picture processing by computer complex and recognition of human faces. Technical Report, Kyoto University Department of Information Science
15. Kullback S, Liebler RA (1951) On information and sufficiency. Ann Math Stat 22(1):79–86
16. Kuniyoshi Y, Kita N, Rougeaux S, Suchiro T (1995) Active stereo vision system with foveated wide angle lenses. In: Asian conference on computer vision
17. Loy G, Fletcher L, Apostoloff N, Zelinsky A (2002) An adaptive fusion architecture for target tracking. In: 5th international conference on automatic face and gesture recognition
18. Matuszyk L (2001) Tracking with the Cable Drive Active Vision Robot. A final-year honours thesis, Australian National University, Canberra
19. Murray WW, Du F, McLauchlan PF, Reid ID, Sharkey PM, Brady M (1992) Active Vision, pp 155–172
20. Pahlavan K, Eklundh JO (1992) A head-eye system – analysis and design. CVGIP: Image Understanding: Special issue on purposive, qualitative and active vision
21. Sharkey PM, Murray DW, Heuring JJ (1997) On the kinematics of robot heads. IEEE Trans Robot Automat 437–444
22. Soto A, Khosla P (2001) Probabilistic adaptive agent based system for dynamic state estimation using multiple visual cues. In: Proc. international symposium of robotics research (ISRR)
23. Suzuki A, Yasui N, Nakano N, Kaneko M (1992) Lane recognition system for guiding of autonomous vehicle. In: Proc. symposium on intelligent vehicles, pp 196–201
24. Thrun S (2000) Probabilistic algorithms in robotics. AI Mag 21(4):93–109
25. Triesh J, von der Malsburg C (2000) Self-organized integration of adaptive visual cues for face tracking. In: Proc. IEEE international conference on face and gesture recognition, pp 102–107
26. Williamson T, Thorpe C (1999) A trinocular stereo system for high-way obstacle detection. In: Proc. international conference on robotics and automation (ICRA99)
27. Wilson HR, Cowan JD (1972) Excitatory and inhibitory interactions in localized populations of model neurons. Biophys J 12:1–24

Andrew Dankers is currently enrolled in the Master of Philosophy course at the Australian National University's Research School of Information Sciences and Engineering Robotics System Laboratory, where he works on the vision processing systems for the Autonomous Vehicle project led by Professor Zelinsky. He has obtained a degree in science specialising in physics (1999) and a degree in systems engineering (2003) and worked as a robotics research assistant at the same institution. In 2004, Andrew undertook a research internship with the Department of Humanoid Robotics and Computational Neuroscience at the Advanced Telecommunications Research Institute in Japan. His current research interests are in computer vision and mobile robotics for real-world interactions.

Alexander Zelinsky received his Ph.D. in robotics in 1991. He worked for BHP Information Technology as a computer systems engineer for 6 years before joining the University of Wollongong, Department of Computer Science, as a lecturer in 1984. Since joining Wollongong University, he has been an active researcher in the robotics field. Dr Zelinsky spent nearly 3 years (1992–1995) working in Japan as a research scientist with Professor Shinichi Yuta at Tsukuba University and Dr Yasuo Kuniyoshi at the Electrotechnical Laboratory. In March 1995 he returned to the University of Wollongong, Department of Computer Science, as a senior lecturer. In October 1996, Dr Zelinsky joined the Australian National UNiversity, Research School of Information Sciences and Engineering, as head of the Robotic Systems Laboratory. In January 2000 Dr Zelinsky was promoted to professor and head of systems engineering at the Australian National University. In the same year Professor Zelinsky founded a computer vision company called Seeing Machines. Professor Zelinsky is a member of the IEEE and the IEEE Computer Society and was president of the Australian Robotics & Automation Association (1998–2000).