Research Report 213

An Heterogeneous M-SIMD Architecture for Kalman Filter Controlled Processing of Image Sequences

T.J. Atherton, D.J. Kerbyson and G.R. Nudd

RR213

An heterogeneous Multiple-SIMD (M-SIMD) architecture is used to analyse image sequences by integrating image processing operations with optimal recursive (Kalman) estimators. The architecture uses SIMD processors for data parallel (iconic) operations and MIMD processors for control parallel (numeric and symbolic) tasks. The SIMD processors are configured as small contiguous sub-arrays, with each subarray attached to a single MIMD processor. This allows operational autonomy intermediate between the pure data and control parallel paradigms. Use of the architecture is illustrated with size-based detection and segmentation techniques that are guided by Kalman filters through an image sequence. The measurements of object size on the image plane allow a robust estimate of the range and physical size of the object in the world. Results are presented for the architecture performance, the image processing, and the estimation techniques.

Department of Computer Science
University of Warwick
Coventry CV4 7AL
United Kingdom

December 1992

An Heterogeneous M-SIMD Architecture for Kalman Filter Controlled Processing of Image Sequences

T.J. Atherton, D.J. Kerbyson, and G.R. Nudd

Department of Computer Science, University of Warwick, Coventry, CV4 7AL, UK, and, WSTL, University of Warwick Science Park, Coventry, UK.

Abstract

An heterogeneous Multiple-SIMD (M-SIMD) architecture is used to analyse image sequences by integrating image processing operations with optimal recursive (Kalman) estimators. The architecture uses SIMD processors for data parallel (iconic) operations and MIMD processors for control parallel (numeric and symbolic) tasks. The SIMD processors are configured as small contiguous sub-arrays, with each sub-array attached to a single MIMD processor. This allows operational autonomy intermediate between the pure data and control parallel paradigms. Use of the architecture is illustrated with size-based detection and segmentation techniques that are guided by Kalman filters through an image sequence. The measurements of object size on the image plane allow a robust estimate of the range and physical size of the object in the world. Results are presented for the architecture performance, the image processing, and the estimation techniques.

1 Introduction

Parallel array architectures are typically classified as SIMD (e.g. DAP, Connection Machine [1,2]) or MIMD (iWarp, Transputer). SIMD architectures are suited to data parallel problems where all elements, in a large array of data, are processed identically, and MIMD architectures are effective where a number of data elements are to be processed independently (control parallel). When information must be extracted about several independent objects present in an image sequence both types of parallelism are required.

The issue of control in image processing and image understanding has been investigated widely in terms of bottom-up (data driven) approaches, top-down (model driven) approaches, and also in terms of approaches intermediate between these two extremes [3]. Knowledge based techniques [4, 5], through to approaches using optimal least squares estimators [6] have been used to control the processing of single images. We are interested here in processing sequences of images. Optimal recursive estimators are well suited to image sequence processing and have been used in a number of application areas [7, 8, 9, 10]. Each image of a sequence contains insufficient information for the robust control of the operations on that image. However, the information contained in each noisy image measurement may be integrated in an optimal way through an image sequence. Image processing operations provide the measurements from the image data that are the inputs to the estimators. The resulting estimations are then used to control the image processing operations on the next image of the sequence:

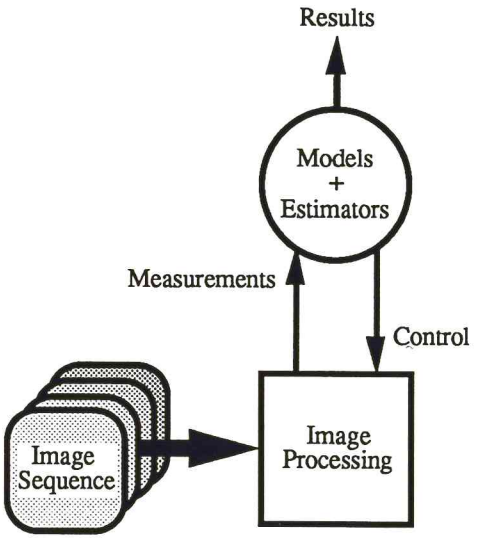

Figure 1. Overview of the processing scheme.

We illustrate this approach with an object tracking and range estimation problem. A typical requirement, before performing higher level operations, is to detect objects of a particular class within each image and then to segment these objects from their backgrounds. The processes of detection and segmentation are, in general, improved by increasing the a priori knowledge of the imaged object [11, 12, 13]. A priori information about the image of an object, as it appears in a sequence of images, is contained within models and updated by optimal recursive estimators. The initial a priori information takes two forms. Models of the imaged object and its dynamic behaviour, together with uncertain estimates of the quantities required to control the image processing. As each image is processed the information contained in the measurements is used to reduce this uncertainty, i.e. increase the a priori information (in Fisher's sense) in the estimates that control the processing of the next image in the sequence.

We will show that two types of computation dominate the task. The processing of images is data parallel; the early stages of the image processing computations are iconic, global, and use small integer word sizes. Later stages of the image processing are iconic but local to objects. The processing may be guided by estimating properties of the objects. This estimation is a numeric and control parallel problem if multiple objects and multiple models are allowed. The numeric computations are floating point intensive. The two extreme types of computation map onto two classes of parallel architecture. Data parallel Single Instruction Multiple Data (SIMD) for the iconic processing, and Multiple Instruction Multiple Data (MIMD) for the control parallel numeric processing.

This diversity of processing requirements leads to a difficulty of parallelising the overall task on one class of architecture. Amdahl's Law [14] tells us that even small parts of programs must be parallelised to achieve linear speedup with increasing numbers of processors. If P is the fraction that is parallelisable and N the number of processors then:

$$speedup = \left[(1-P) + \frac{P}{N} \right]^{-1} \qquad (1)$$

As N increases, this asymptotically approaches $(1\text{-}P)^{-1}$. By using only one form of processor, either data or control parallel, the fraction (1-P) may remain significant at the task level. The approach of the heterogeneous M-SIMD architecture is to provide compute capability to tackle both forms of computation simultaneously, i.e. to reduce

the serial component (1-P). If the fraction of the computation that is data parallel is D, and the fraction that is control parallel is C, then combining these gives a modified form of Amdahl's Law:

$$speedup = \left[(1-(D+C)) + \frac{D}{N_{SIMD}} + \frac{C}{N_{MIMD}} \right]^{-1} \quad (2)$$

The combination of SIMD and MIMD parallel arrays further reduces the fraction (1 - (D + C)) increasing the available speedup. By partitioning the SIMD array into subarrays with operational autonomy, that is creating processors with granularity and characteristics intermediate between SIMD and MIMD, to give an M-SIMD architecture, we can introduce further parallelism with the proportion, G, that can be processed in this way:

$$speedup = \left[(1-(D+C+G)) + \frac{D}{N_{SIMD}} + \frac{C}{N_{MIMD}} + \frac{G}{N_{MSIMD}} \right]^{-1} \quad (3)$$

The speedup is now limited by the remaining serial fraction of the original code and asymptotically approaches $[1-(D+C+G)]^{-1}$. The use of multiple subarrays independently in this way can be thought of as an array of MIMD processors each with a two-dimensional ALU, data path, and registers.

We describe an architecture that combines the properties of bit-serial SIMD processors with bit-parallel floating point MIMD processors. We add a further refinement to this combination of processor types, at an additional hardware cost of ~10%, which allows limited operational autonomy [15] within the SIMD array by dividing it into a number of subarrays:

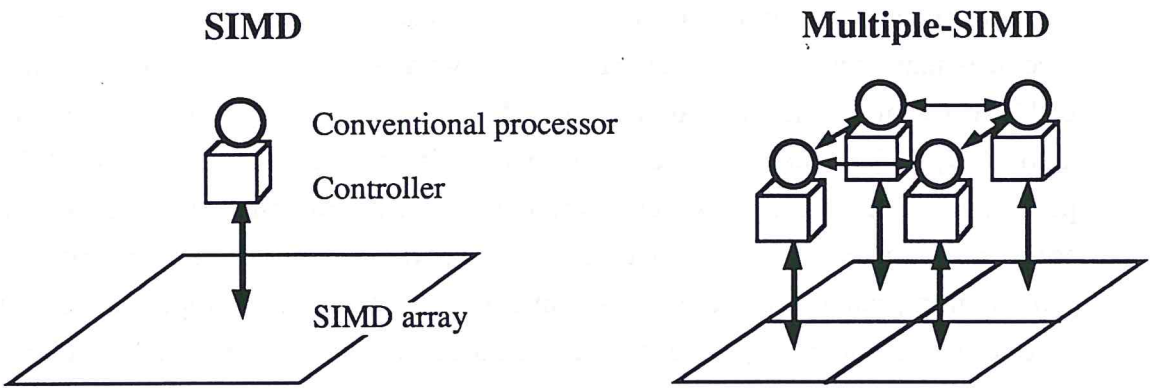

Figure 2. Comparison of SIMD and M-SIMD architectures.

Tracking the position and size of approaching objects through an image sequence is used to illustrate the mapping onto the architecture. The image size of a stationary automobile, as it is approached by a camera, is used to estimate the distance to the vehicle [16]. The image processing operations that detect the object and segment it from its background are sized based. The operations are first guided to detect objects of an approximate size [17,18]. False detections are eliminated by comparing measured positions with predicted positions [19]. The object is segmented from its background using boundary points of the object given a prediction of the objects size [18, 20, 21]. The recursive estimators maintain and refine estimates of both position and size which are used to guide the image processing. The example reported here could be extended to include more complex models of objects, and to distinguish between multiple objects by matching against more than one model and performing a consistency analysis [19]. Optimal recursive estimators provide a robust and potentially powerful tool for the control of image sequence analysis.

2 Architecture

We have designed, built, and tested a prototype heterogeneous Multiple-SIMD architecture. The architecture consists of an SIMD array divided into subarrays each with an individual control stream and an associated conventional processor. Each subarray, controller and conventional processor is referred to as a cluster. An implementation of the architecture may consist of one or more clusters. The Warwick Pyramid Machine consists of an 8 by 8 array of clusters.

The clusters may operate individually or form groups of two or more and run synchronously. Several of these groups of clusters may then operate concurrently.

The operational autonomy of the clusters differentiates the architecture from machines that simply combine an SIMD array with an MIMD array. The advantages of M-SIMD are greater utilisation of the SIMD processors (by allowing operational autonomy) and increased vertical data bandwidth between the SIMD and MIMD processors. The cost is increased complexity both in software and hardware. The software complexity is difficult to quantify but raises interesting issues of mixed programming paradigms, and load-balancing. The hardware complexity is of the order of an additional 10% over a machine consisting of a conventional SIMD array combined with a conventional MIMD array.

Figure 3 shows a schematic of four clusters. Each of the prototype clusters is implemented as a 16 by 16 SIMD array of bit-serial processors, with each processor having 8K bits of memory. The instruction stream to this array is provided by a 16-bit microcoded controller, which has 16K of 64 bit horizontal microcode memory and 4K of 16 bit dual-port memory. This dual-port memory is for communication of program and data to the MIMD processor. The MIMD processors currently used are Transputers [22]. The prototype uses DAP SIMD Processing Elements (PE's) [1].

The bit-serial SIMD PE's are four-way nearest neighbor connected for data transfer. The controllers are four-way nearest neighbor connected for control and synchronisation, and the MIMD processors are four-way connected (via 20 Mbit s^{-1} Inmos bit-serial links) for data communication.

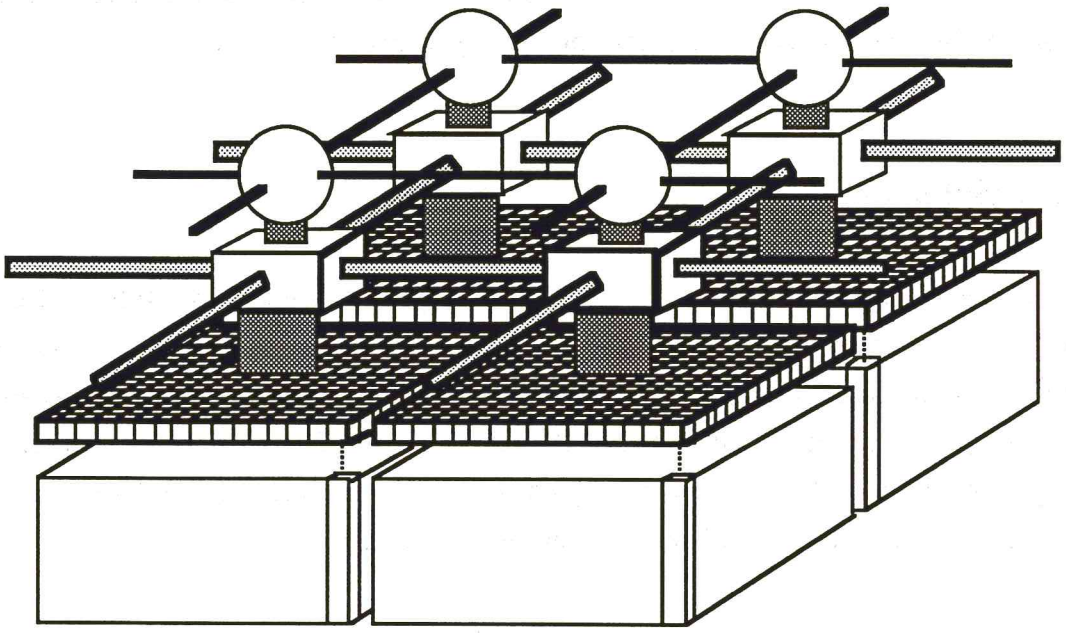

Figure 3. Four clusters of an M-SIMD architecture.

3 Processing Flow

The processing flow that we describe is typical of the enhancement, detection, segmentation, feature extraction and classification approach for each frame in an image sequence. The early stages are global iconic filtering operations. The later stages of the processing are local to individual objects, and progressively less iconic and more numeric. We describe the global detection and local segmentation stages of this image processing in detail. Both the detection and segmentation techniques require an estimate of the objects image size as input. The image size and position of each object are tracked using (in this example) optimal recursive estimators in the form of Kalman filters.

Objects are first detected within an image and then segmented to produce a measure of object size. This may be carried forward to guide the detection and segmentation in the subsequent frame. Image size may change in a systematic way from one image to the next. A model of the way size changes with time may be used to integrate size measurements within a tracking filter. From the current measurement the filter produces a prediction of the size in the next frame, and an associated uncertainty. This size estimate is used to control the image processing operations applied to the next frame. Estimates of the objects physical size and actual range from the camera image plane may also be computed.

The objects position on the image plane is tracked, giving a prediction of where the object will occur in the next frame. This serves two purposes. It limits the processing in the next image to a region of interest containing the object, and the tracked centroid is used to select the true object out of possibly many false detections in the next frame.

The process of integrating image measurements with Kalman filters is shown in Figure 4. The first operation on the n^{th} image is a global edge orientation detection followed by a global detection of objects whose predicted size, $\Delta\mathbf{x}(n|n-1)$, is available from previous frames. The set of detected objects have centroids $\{\mathbf{x}_d(n)\}$. These are compared with the predicted position of the object, $\mathbf{x}(n|n-1)$, by a data association operation that selects one detection, $\mathbf{x}_d(n)$, as the detected centroid of the object being tracked by this filter. Using this centroid and the same size prediction as used for the detection, $\Delta\mathbf{x}(n|n-1)$, the segmentation operation separates the pixels within the object

from those of the background. From the segmentation a centroid, $\mathbf{x}_S(n)$, and size, $\Delta\mathbf{x}_S(n)$, are measured. These measurements are used by the tracking filters to produce predictions of the objects location and size in the next frame. Both filters are recursive optimal estimators, Kalman filters [23].

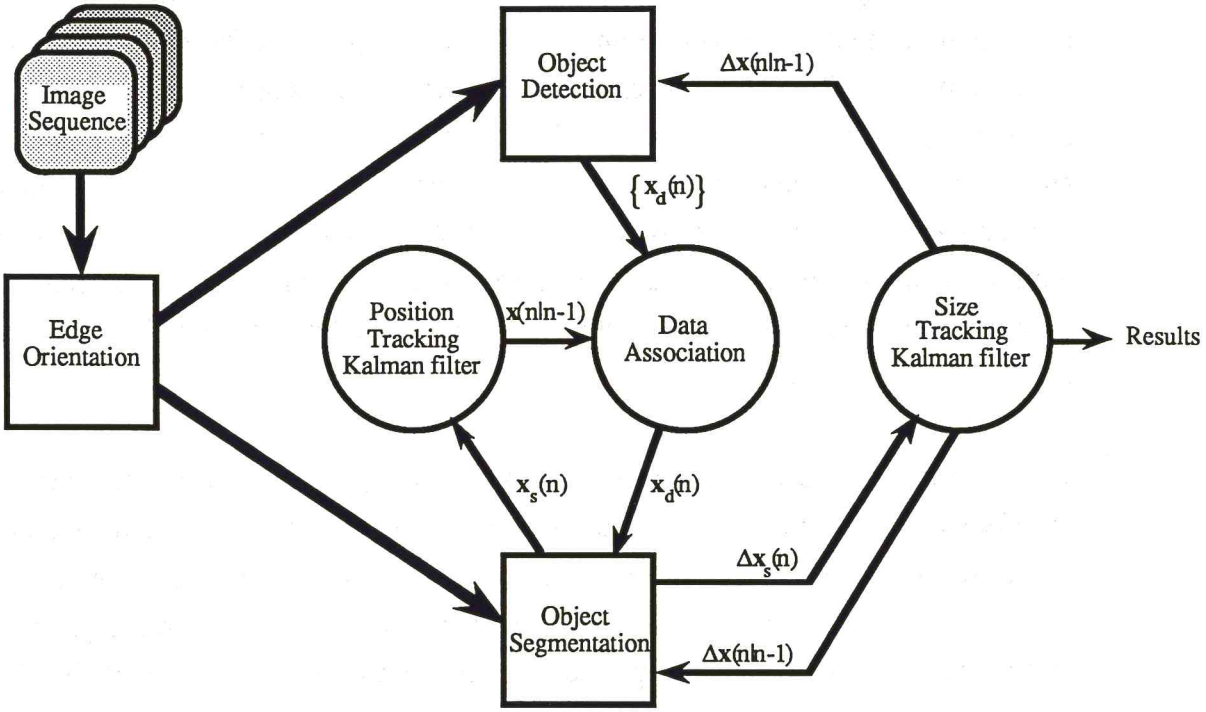

Figure 4. Integration of image processing and optimal estimation using size and position measurements.

Uncertainties about the image plane measurements are incorporated into the Kalman filters, but are not shown in Figure 4. The sources of measurement noise include, the segmentation (such as an object in a cluttered scene) for the object size measurement, and unsteady camera motion for the centroid tracking. The Kalman filters undergo a bootstrap operation when an object is first detected.

4 Image Processing

The sized based detection and segmentation are described below. Initially, each image undergoes an enhancement stage, and an edge detection stage that gives an approximate edge orientation and magnitude across the image.

4.1 Detection

The spoke filter is used for the detection of objects, using edge magnitude and orientation information [17]. It detects objects which lie within a specified range of sizes, and which are approximately convex in shape. It is a refinement of the circle Hough transform [24] with a particularly efficient implementation on bit-serial SIMD parallel array architectures [18]. It can detect 'blobs' which are not exactly, but only approximately, circular and convex in shape.

The spoke filter [17] has as input an edge detected image, $\mathbf{O}(\mathbf{x})$. Edge orientation is quantised to eight directions and the edge magnitude is thresholded at a magnitude of t, to retain the top T percent of edge points to give the image $\mathbf{T}(\mathbf{x})$:

$$\mathbf{T}(\mathbf{x}) = \begin{cases} \mathbf{n}_i & ; \|\mathbf{O}(\mathbf{x})\| \geq t, \mathbf{O}^{\dagger}(\mathbf{x})\mathbf{n}_i > \mathbf{O}^{\dagger}(\mathbf{x})\mathbf{n}_j, i, j \in [0,7], j \neq i \\ \mathbf{0} & ; \textit{otherwise} \end{cases} \quad (4)$$

where the unit vector $\mathbf{n}_i = \begin{bmatrix} \cos\theta_i \\ \sin\theta_i \end{bmatrix}$, $\mathbf{O}^{\dagger}(\mathbf{x})$ is the transpose of the vector of partial spatial derivatives of the image intensity at position $\mathbf{x}$, and the discrete orientations are $\theta_i = \frac{\pi i}{4}$.

The edge points, in $\mathbf{T}(\mathbf{x})$, are projected in the direction of their orientation, from a distance R_{min} to R_{max} from the edge point to form eight binary spoke register images, $R_i(\mathbf{x})$, one for each orientation as described above:

$$\begin{aligned} R_i(\mathbf{x}) &= 1 \quad ; \quad K > 0 \\ R_i(\mathbf{x}) &= 0 \quad ; \quad \text{otherwise} \end{aligned} \quad (5)$$

where $i \in [0, 7]$, $\mathbf{n}_i$ is given in equation 4, and:

$$K = \text{card}\left\{\mathbf{T}(\mathbf{w}) \;\middle|\; \|\mathbf{T}(\mathbf{w})\| > 0, \left\|\left[\mathbf{I} - \mathbf{n}_i\mathbf{n}_i^{\dagger}\right](\mathbf{w}-\mathbf{x})\right\| < 1, R_{\min} \le \|(\mathbf{w}-\mathbf{x})\| \le R_{\max}\right\}$$

R_{min} and R_{max} define the range of sizes of an object that the filter will detect. The i^{th} spoke register for each image position, **x**, will have a 1 present if the edge magnitude at any point, **w**, within a distance R_{min} to R_{max} exceeded the threshold t, and if the quantised edge orientation **T**(**w**) is parallel (or anti-parallel) to the vector from **x** to **w**. The values of R_{min} and R_{max} can be chosen well if the size of the object in the image is known a priori. The spoke register images are smoothed, to give $R_1^*(\mathbf{x})$, over a 3 by 3 neighborhood around each **x**, $\Re(\mathbf{x})$:

$$R_i^*(\mathbf{x}) = \bigcup_{u \in \Re(\mathbf{x})} R_i(\mathbf{x}) \tag{6}$$

This overcomes the possibility of different spokes crossing without intersecting (a problem associated with the image sampling). Edge orientations point towards the centre of convex objects lighter than their backgrounds. There is a peak in the number of spoke intersections near the centre of the object. The spoke intersection image, F(**x**), gives a count of the number of spokes crossing at each position:

$$F(\mathbf{x}) = card\left\{R_i^*(\mathbf{x}), i \in [0,7]\right\} \tag{7}$$

The centroids of regions of 7 or 8 spoke intersections are the detected positions. When the contrast of the object is not solely lighter (nor darker) than its background, or is not known, spokes radiate in the direction of the edge orientation and the edge orientation + π.

Different spoke lengths (R_{min} and R_{max}) for each edge orientation detect objects of unequal width and height. The width and the height of the object were modelled in the processing flow described in this report. Other spoke lengths were elliptically interpolated. The Kalman filter gives the predicted size in the current frame, $\Delta x(n|n-1)$. R_{min} is 20% less than this value and R_{max} 20% more.

Figure 5 shows a simplified example. Only eight edge points of an object are shown for simplicity (one for each of the orientations). The width of spokes is 3 pixels, R_{min}=5, and R_{max}=13 pixels.

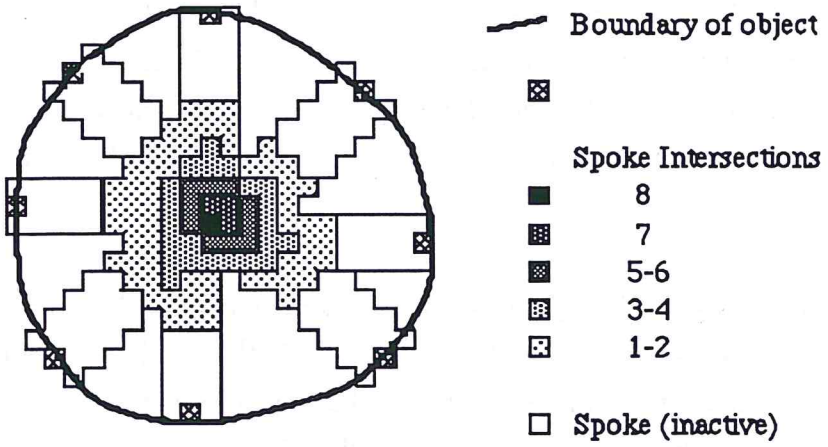

Figure 5. Example of the Spoke filter

4.2 Segmentation

After detection and identification of the tracked object from the many possible detections (see Section 5.1), a technique we have called the "Spoke Segmentor" [18] segments the object from its background. Similar techniques reported by [20, 21] are all closely related to the wedge filter approach [25]. In a similar way to the spoke detection filter, the spoke segmentor operates on objects that are approximately convex in shape and lie within a range of sizes.

Spokes radiate out across the edge image from the detection centroid, $\mathbf{x}_d(n)$, at angles of $2\pi/M$ (where M = number of segmentation spokes). Spokes are active from some distance r_{min} to r_{max} from the centroid. A maximum likelihood, weighted, peak edge magnitude, along the active spoke region is the object boundary contour point:

$$\mathbf{b}_i(n) = \arg\max_{\mathbf{w}} \left\{ N\left(\left\| \mathbf{w} - \left(\mathbf{x}_d(n) + \frac{\Delta x_i(n|n-1)}{2} \mathbf{m}_i \right) \right\|, p_{\Delta x_i}(n|n-1) \right) \left\| \mathbf{O}^{\dagger}(\mathbf{x}) \mathbf{m}_i \right\| \mid W \right\} \tag{8}$$

where $\mathbf{w} \in W$, the set of points on the spoke, and:

$$W = \left\{ \mathbf{w} \;\middle|\; \left\| \left[\mathbf{I} - \mathbf{m}_i \mathbf{m}_i^{\dagger} \right] (\mathbf{w} - \mathbf{x}_d(n)) \right\| < 1, r_{\min} \le \left\| \mathbf{w} - \mathbf{x}_d(n) \right\| \le r_{\max} \right\}$$

$\mathbf{m}_i$ is the unit vector in the i^{th} spoke direction, $N(x, \sigma^2)$ is the normal distribution, and $p_{\Delta x_i}(n|n\text{-}1)$ is the variance in the predicted size in the i^{th} orientation (available from the predicted covariance matrix of the size tracking filter, the imaging model, and the predicted size estimate). Note that the edge magnitudes along each spoke are further weighted by the dot product between the local edge orientation and the direction of the radiating spoke.

Figure 4 shows the signal flow graph for the spoke segmentor. The result of this operation is a set of boundary points, $\{\mathbf{b}_i(n) \mid i \in [0, M\text{-}1]\}$. In high clutter situations some of these points may not lie on the object boundary but instead on other edge features. Post-processing overcomes this. The collated boundary points form a 1D histogram of boundary distance from the detected centroid, $\|\mathbf{b}_i(n) - \mathbf{x}_d(n)\|$. A median filter removes isolated rogue spikes from the 1D histogram of boundary distances.

A convex hull operation performed on the resulting boundary points produces an approximate boundary contour. A measure of size, $\Delta\mathbf{x}_S(n)$ and the segmentation centroid, $\mathbf{x}_S(n)$, are extracted from this image region. The pixels extracted from the convex hull are used to classify the object - we do not discuss this further here.

We have found that by tracking both the object width and height a reliable detection and segmentation can be achieved through an image sequence.

5 Integration of image sequence measurements

Measurements, with associated noise variances, incorporated into a Kalman filter produce estimates of $\mathbf{s}(n)$ - the system state and $\mathbf{P}(n)$ - the covariance of the state. The plant noise describes the error between the system model and the true system. Measurement noise describes the error between the measurement and the true observable. The filter is recursive producing estimates of the state, with associated covariances, as well as predictions for the next measurement. The state estimates are updated with each new measurement by one iteration of the Kalman filter [19]. Two filters are required for each object, one for the centroid tracking and the other for the size estimation.

5.1 Centroid Tracking

Tracking is the modelling of an object's kinematic state by the processing of measurements obtained from sensors. The measurements are noise-corrupted observations used to update the estimates of the state of the observed system. Ambiguity exists in associating measurements with tracked objects in high density situations, i.e. which measurement should be associated with which tracked object. Several methods exist to solve this data association problem in situations of both low and high densities of objects, including probabilistic and Bayesian approaches [19].

We use position and velocity tracking in both the x and y dimensions of the image. Image plane co-ordinates are used throughout. The system model for this is:

$$\mathbf{s}_x(n) = \mathbf{A}_x \mathbf{s}_x(n-1) + \begin{bmatrix} 1/2 \\ 1 \\ 1/2 \\ 1 \end{bmatrix} w_x(n) \tag{9}$$

where

$\mathbf{A}_x = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$, $\mathbf{s}_x(n) = [x \quad \dot{x} \quad y \quad \dot{y}]^\dagger$, and $w_x(n)$ is the plant noise with variance $\sigma^2_{w_x}$ representing inadequacies in the kinematic model. (the subscript **x** indicates that we are dealing with quantities from the position tracking filter.) The measurement model is:

$$\mathbf{x}(n) = \mathbf{C}_x \mathbf{s}_x(n) + \begin{bmatrix} 1 \\ 1 \end{bmatrix} \mathbf{v}_x(n) \tag{10}$$

where $\mathbf{C}_x = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$, $\mathbf{x}(n) = [x \quad y]^\dagger$, and $\mathbf{v_x}(n)$ is the measurement noise with variance σ^2_{vx}.

The prediction of the centroid in the next frame is:

$$\mathbf{x}(n+1|n) = C_x A_x \mathbf{s}_x(n|n) \tag{11}$$

Standard Kalman filter techniques give the state and covariance updates and predictions. The filter undergoes a bootstrap process such that the centroid of the first segmentation, with measurement variance σ_{vx}^2, is used as the initial state and initial covariance matrix $P_x(0)$.

In this application we assume a low density of objects in each image of the sequence. The Nearest-Neighbor Standard Filter (NNSF) data association technique is then appropriate [19]. In the NNSF the measurement nearest to the prediction from the kinematic system is used. The Mahalanobis distance [26], between the predicted centroid position, $\mathbf{x}(n|n-1)$, and the detection measurements $\{\mathbf{x}_d(n)\}$, is computed, as:

$$d^2(\mathbf{x}) = \left[\mathbf{x}_d(n) - \mathbf{x}(n|n-1)\right]^{\dagger} \mathbf{P}^{-1}(n|n-1)\left[\mathbf{x}_d(n) - \mathbf{x}(n|n-1)\right] \quad (12)$$

The match in the n^{th} image with the minimum Mahalanobis distance corresponds to the tracked object. The NNSF determines which of the detections from the spoke filter corresponds to the tracked object in the current frame. The spoke segmentor uses the centroid of this detected region that has passed the NNSF test. The measured centroid of the segmented object, $\mathbf{x}_S(n)$, updates the position tracking filter. In the images used below this is sufficient for the association of objects between image frames.

5.2.1 Size tracking and range estimation

Figure 6 shows the image size of an object on the image plane. By similar triangles:

$$\Delta x(n) = \frac{\Delta X}{Z(n)} fk = \frac{\Delta X}{Z_0 - S(n)} fk \quad (13)$$

Where the range to the object in frame n, Z(n) metres, is the initial range, Z_0, minus the distance travelled towards the object, S(n). The physical size of the object is ΔX metres, the object size on the image plane (in pixels) is $\Delta x(n)$, k is a constant relating pixel distances to metres and f is the focal length of the camera in metres. Using the inverse image size the relationship between the image size of the object and the range to the object is linearised [16]:

$$y(n) = \frac{1}{\Delta x(n)} = \frac{Z_0}{kf\Delta X} - \frac{S(n)}{kf\Delta X} = a + bS(n) \quad (14)$$

The constants a and b are the quantities to be estimated by the size tracking filter. This relationship describes the way the size develops as a camera approaches an object and allows the range to the object and physical size of the object to be determined from the estimates of a and b.

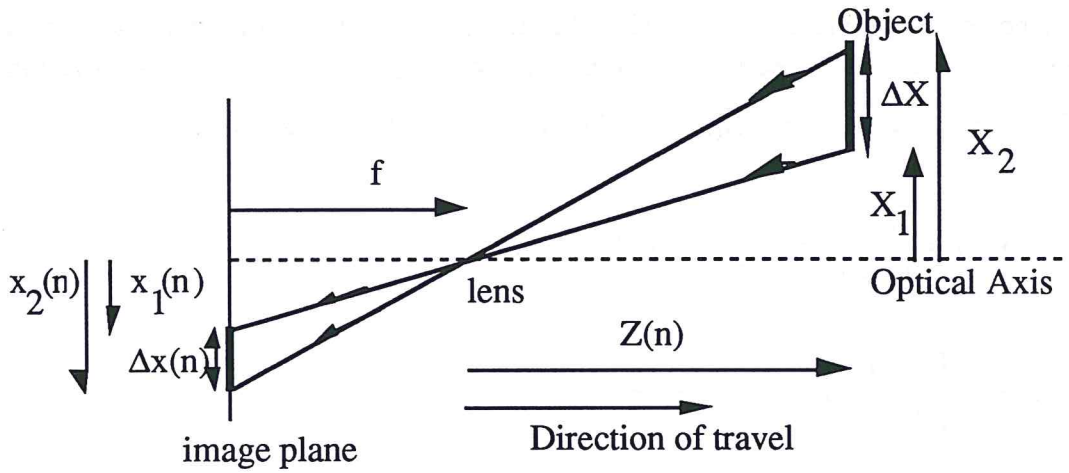

Figure 6. The Imager

The use of relative position (size) removes the need to find the focus of expansion, making the technique immune to small random rotations of the translating camera. The variation in depth of the occluding boundary is assumed small with respect to the distance from the camera (see Discussion section 7).

5.2.2 Image Size Measurements

A first order error analysis gives the variance in the inverse image size measurements, $p_{yy}(n)$, in terms of the variance of the true image size measurements, $p_{\Delta\Delta}$:

$$p_{yy}(n) \approx \frac{p_{\Delta\Delta}}{\Delta x^4(n)} = p_{\Delta\Delta} y^4(n) = \sigma_{v_y}^2(n) \qquad (15)$$

This estimate of the variance in the inverse image size varies inversely with the fourth power of the true image size. Assuming $p_{\Delta\Delta}$ is time invariant (which is a reasonable model for the spoke segmentor), $p_{yy}(n)$ reduces rapidly as the object size increases.

5.2.3 Inverse Size Kalman Filter Formulation

Two states model the system, equation 16. It has a single input y(n), the inverse image size measurement, equation 17. The additive plant noise process, $w_y(n)$, is approximated by zero mean Gaussian noise with variance $\sigma^2_{w_y}(n)$ representing the plant noise in the inverse image size due to uncertainties in the camera forward motion. The additive measurement noise process, $v_y(n)$, has variance $\sigma^2_{v_y}(n)$ that varies with true image size (equation 11). The state evolution model is:

$$\mathbf{s}_y(n) = \begin{bmatrix} a(n) \\ b(n) \end{bmatrix} = \mathbf{A}_y \mathbf{s}_y(n-1) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} w_y(n) \tag{16}$$

where $\mathbf{A}_y = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$. The measurement model is:

$$y(n) = \begin{bmatrix} 1 & S(n) \end{bmatrix} \begin{bmatrix} a(n) \\ b(n) \end{bmatrix} + v_y(n) = \mathbf{C}_y(n)\mathbf{s}_y(n) + v_y(n) \tag{17}$$

The predicted inverse size for the next time frame is given by:

$$y(n+1|n) = \begin{bmatrix} 1 & S(n+1|n) \end{bmatrix} \begin{bmatrix} a(n+1|n) \\ b(n+1|n) \end{bmatrix} \tag{18}$$

i.e. using the current estimates of the states a and b, and the distance that will have been travelled when the next frame is acquired. Standard Kalman filter techniques give the state and covariance updates and predictions.

5.2.4 Initial conditions for the inverse size Kalman filter

The inverse image size Kalman filter requires an initial estimate of both the states, $\mathbf{s}_y(0)$, and the covariance matrix, $P_y(0)$, which reflects the uncertainty in the initial state. The state is given by $\mathbf{s} = \mathbf{f}(\mathbf{z})$, a vector function of a vector independent variable, where $\mathbf{z} = \begin{bmatrix} Z_0 & \Delta X \end{bmatrix}^\dagger$. The covariance matrix for $\mathbf{s}$ in terms of that for $\mathbf{z}$ is then:

$$\mathbf{P}_s = \mathbf{J}_\mathbf{f}^\dagger \mathbf{P}_z \mathbf{J}_\mathbf{f} \tag{19}$$

where $\mathbf{P}_z$ is the covariance matrix for $\mathbf{z}$ and $\mathbf{J}_f$ is the Jacobian of $\mathbf{f}$ with respect to $\mathbf{z}$. $\mathbf{P}_z$ is given by equation 20 where $p_{\Delta X}$ and p_{ZZ} are the variances in the initial estimates of the object physical size and range respectively.

$$\mathbf{P}_z = \begin{bmatrix} p_{\Delta X} & 0 \\ 0 & p_{ZZ} \end{bmatrix} \tag{20}$$

Then combining equations 20, 19, and 14 the initial covariance matrix for the size Kalman filter is given (after some tedious effort) by equation 21 where Z_0, ΔX are the initial estimates of the object range and physical size respectively.

$$\mathbf{P}_y(0) = \left(\frac{Z_0}{kf\Delta X}\right)^2 \begin{bmatrix} \left(\frac{p_{\Delta X}}{\Delta X^2} + \frac{p_{ZZ}}{Z_0^2}\right) & -\frac{p_{\Delta X}}{Z_0 \Delta X^2} \\ -\frac{p_{\Delta X}}{Z_0 \Delta X^2} & \frac{p_{\Delta X}}{Z_0^2 \Delta X^2} \end{bmatrix} \tag{21}$$

This initial covariance matrix is used in the Kalman filter formulation in the usual way [19]. The result given in equation 21 is critical to the successful implementation of the range estimation scheme.

6 Results

The sequence of images was acquired with a Hitachi GP-6ME colour video camera and digitised to a resolution of 360 x 270 pixels. The initial camera distance was 64.5m from a car. The camera was moved forward in steps of 1m, up to a distance of 2.5m from the car making a total of 62 images.

The product of the focal length, f, and the scale factor k (equation 13) was determined by a calibration experiment to be kf = 610. The initial assumptions were: range Z_0 = 100 metres with variance p_{ZZ} = 10,000m^2; ΔX = 2m with variance $p_{\Delta X}$ = 1m^2; image size measurement variance $p_{\Delta x}$ = 25 pixels^2; inverse image size plant noise σ_{wy}^2 = 2*10^{-6}m^{-2}pixels^{-2}. The position tracking Kalman filter had plant noise variance σ_{wx}^2 = 1 pixel^2, and measurement noise variance σ_{vx}^2 = 25 pixel^2. The number of spokes, N_S, used by for the segmentations was 32.

The initial states of the size Kalman filter can be found using equations 13 and 21. The tracking Kalman filter is initialised as discussed in section 5.1 above.

Two of the images from the sequence are shown in Figure 7, at a distance of 22.5m and 6.5m.

Figure 7. segmented images at ranges of 22.5m and 6.5m

The measured object segmentation width is compared with the actual image width in Figure 8 along with the width predications produced by the size Kalman filter. In Figures 9 and 10 the filtered x and y centroid positions are compared with the actual position and measured position on the image plane.

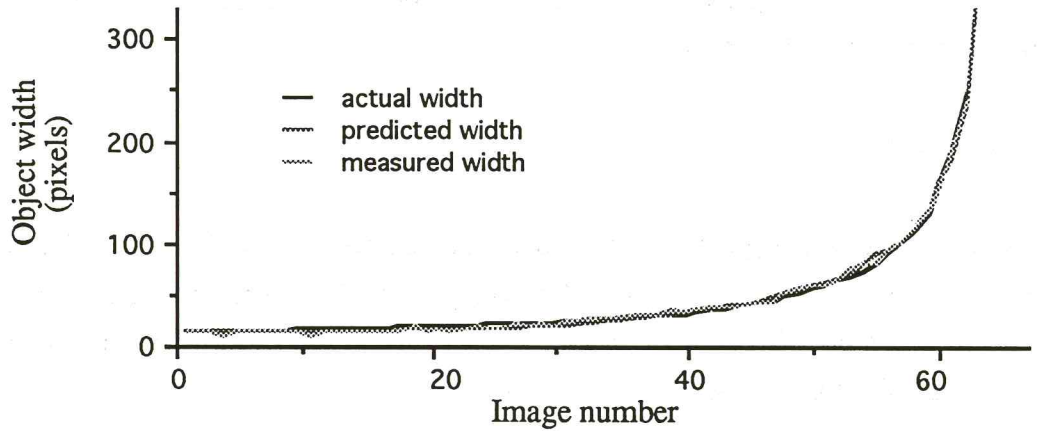

Figure 8. - Image size: actual, measured and predictions

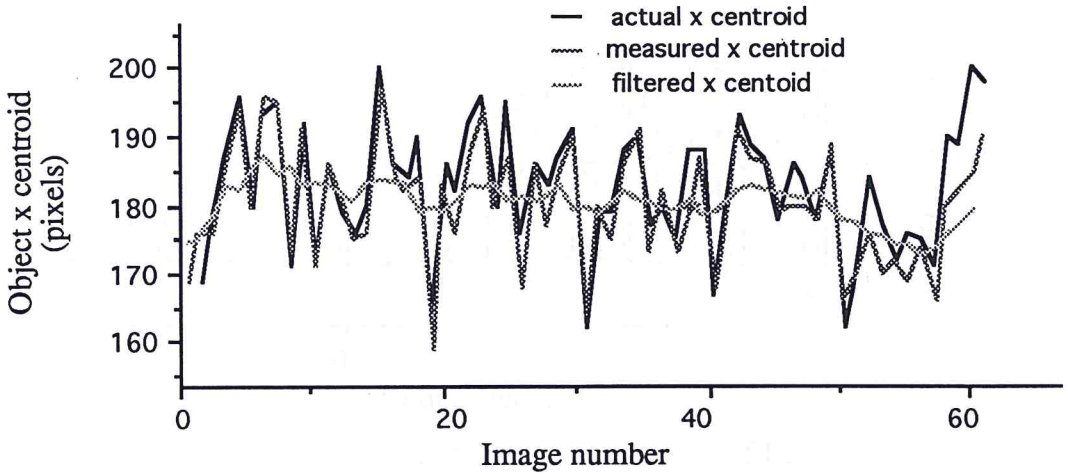

Figure 9. - Object centroid (x), actual, measured and filtered positions

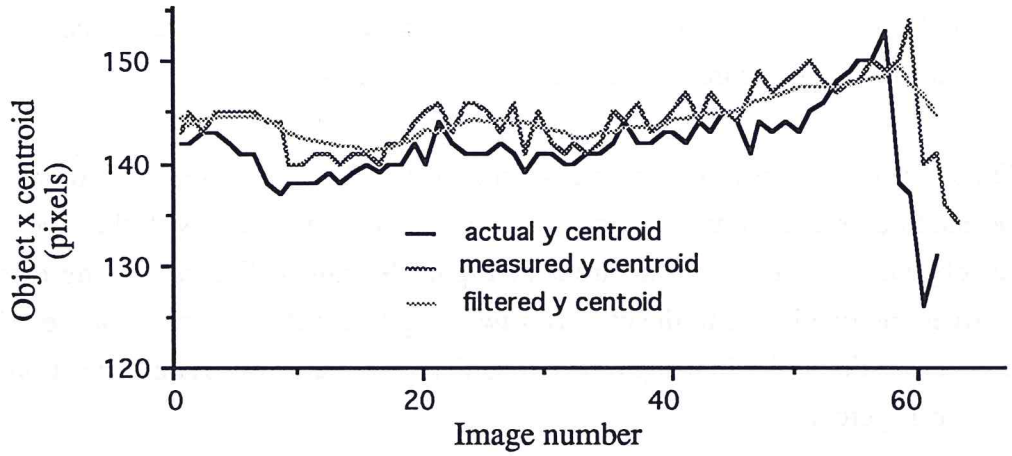

Figure 10. Object centroid (y), actual, measured and filtered positions

The estimates of range and physical size of the object produced by the size Kalman filter are shown in Figures 11 and 12 respectively. The error bars on both are the result of analysis described in section 7 below.

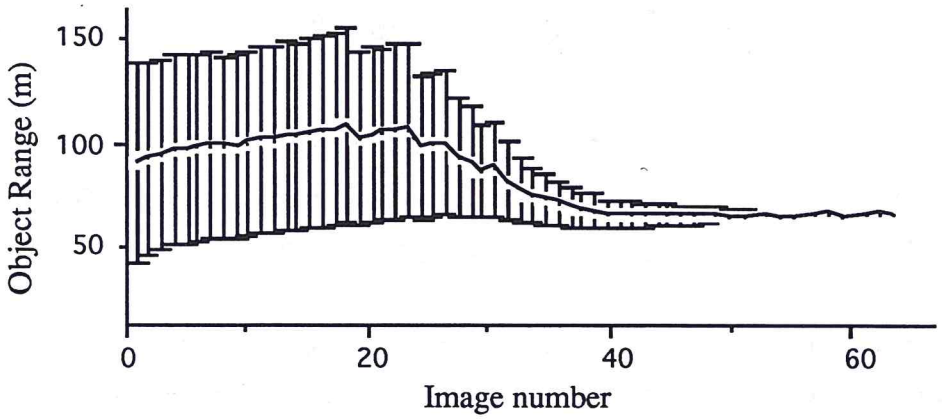

Figure 11. Initial range estimation with error bars

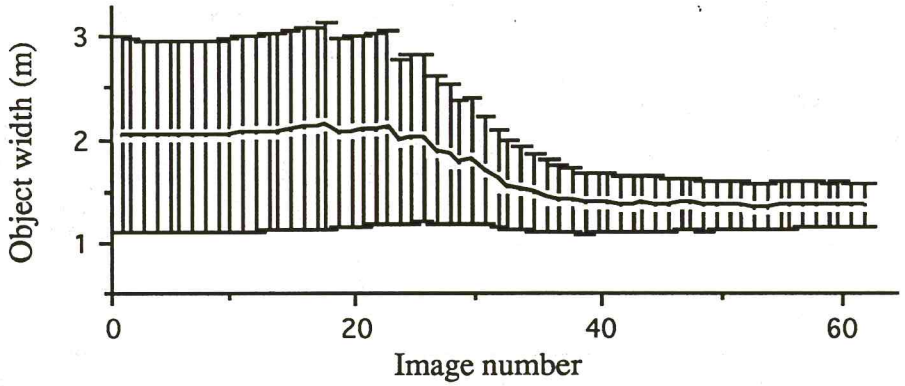

Figure 12. Object physical size estimation with error bars

7 Discussion and Error Analysis

Figure 8 shows the tracking of the object size through the image sequence, the curves show the actual, measured and predicted object widths.

The imager was constrained to move on a plane. This can be seen by comparing the measurements and the tracking of the x centroid of the object, with the y centroid of the object. More noise is apparent in Figure 9 than in Figure 10, the difference between the two is due to the camera yaw being larger than the pitch or the roll. This emphasises the utility in using object (angular) size that is invariant to rotations of the imager, Figure 8.

The uncertainty in the object size and range estimates reduces as the numbers of measurements increase (the error bars in Figures 11 and 12 decrease in size). The variances p_{ZZ} of the estimate of the initial range, Z_0, and $p_{\Delta X}$ of the object physical size, ΔX, using equation 13 can be shown to be:

$$p_{ZZ} \approx Z_0^2\left[\frac{p_{aa}}{a^2}+\frac{p_{bb}}{b^2}-2\frac{p_{ab}}{ab}\right], \quad and \quad p_{XX} \approx \Delta X^2 \frac{p_{bb}}{b^2} \qquad (22)$$

and are used to plot the error bars in Figures 11 and 12. (p_{aa}, p_{bb}, and p_{ab} are the elements of the inverse image size covariance matrix updated from the initial estimates of equation 21 by the inverse image size tracking filter).

The inverse image size state estimate and covariance are shown as a point at the centre of an ellipse, Figure 13. The results shown are for the first image and then every tenth image. Each ellipse represents a locus of points where the probability

density for the system state $\mathbf{s}_y(n)$, using its covariance information $\mathbf{P}_y(n)$, has dropped to 1/e of the maximum. These results are processed to produce estimates of the initial range and object size using equations 23, and the associated errors using equations 22.

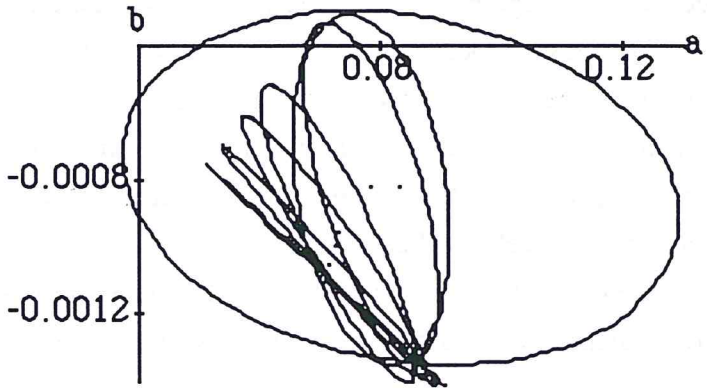

Figure 13 - error ellipses in the a/b space.

The estimated initial range and physical size from equation 14 are given by:

$$Z_0 = -\frac{a}{b}, \quad \text{and} \quad \Delta X = \frac{1}{kfb} \tag{23}$$

The final estimate (smallest ellipse) of initial range was 65.01 ± 0.85m and object physical size was 1.51 ± 0.31m (cf. the true values were 64.50 ± 0.01m and 1.35 ± 0.01m respectively). The true values of the range and size fit within the error bars in Figures 11, 12 respectively, and state estimates lie within the coarse initial estimates, Figure 13.

Extensions to the current approach arise from the observation that the imaged size of the object may change for reasons other than a change in distance between the camera and the object. The object may be rotate and present a different silhouette. This effect may be overcome by adopting a model of the structure of the object [10] and tracking this structure to account for rotations and translations. With a number of models of objects with different internal structure it should be possible to extend the manoeuvre detection techniques from target tracking theory [19] to determine which model is consistent with the measurement data (innovations should be white, zero-mean and compatible with their covariance from the tracking filter). These conditions are met, within a 95% confidence test, for the position and size tracking filters are used here.

8.0 Processing Requirements

We have shown that given a sequence of segmentations of an object the range and size of the object may be estimated. The accuracy of the final range and size estimates for each object in the image plane depends, ceteris paribus, on the accuracy of the segmentation. The image processing techniques used here are generic, the range estimation does not depend on the way that the image size measurements are made, only on its accuracy. Future developments in segmentation techniques can be exploited by the approach outlined here. The algorithms may also vary with the application domain, the level of robust behaviour required of the system, and the time constraints imposed. We have investigated the image processing and estimation processing requirements and their mappings onto a variety of computer architectures.

A comparison of the time taken to perform this processing, for multiple objects, for different architectures, the AMT DAP (Distributed Array Processor) - SIMD, and the Inmos Transputer - MIMD, and the WPM - MSIMD [22], can be seen in Figure 14. For this comparison it is assumed that the size of the SIMD array matches the image size (of 128 x 128 pixels), that the number of MIMD processors matches the number of objects (64) in the image, and that the M-SIMD architecture has 8 by 8 clusters.

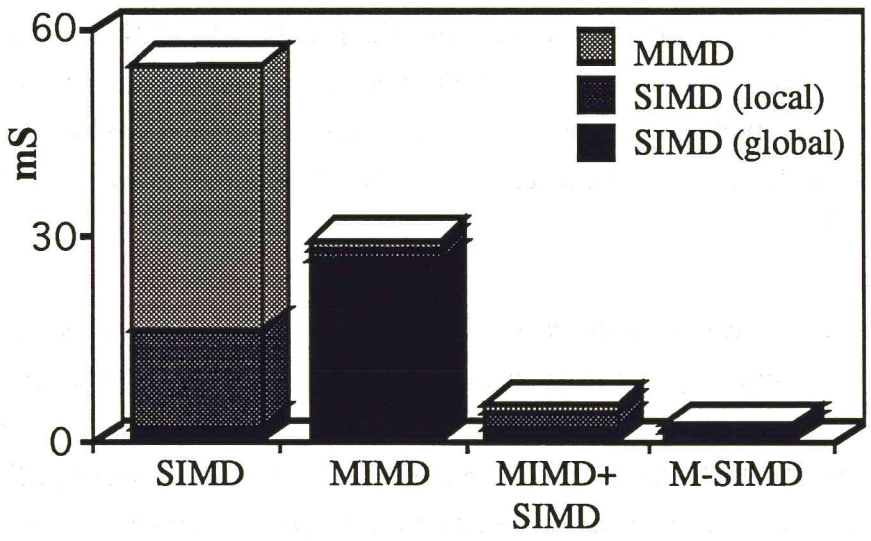

Figure 14. Comparison of the processing time on SIMD, MIMD, MIMD + SIMD, and MSIMD architectures

It can be seen that the M-SIMD architecture out performs SIMD, MIMD, and a direct combination of an SIMD array with an MIMD array. The simple combination of SIMD with MIMD arrays performs poorly on the SIMD local task, which has to be

performed sequentially for each object (it requires local operational autonomy). We are interested in applications where there are large numbers of objects being tracked.

The significant time advantage of the M-SIMD architecture over SIMD and MIMD is achieved at relatively low cost in hardware complexity, see Table 1.

	128 X 128 SIMD	8 X 8 MIMD	SIMD + MIMD	64 Cluster M-SIMD	Additional cost of M-SIMD
Gates	2×10^6	6×10^6	8×10^6	8.6×10^6	6.4×10^5
Memory (bits)	1×10^9	1×10^9	2×10^9	2.7×10^9	6.7×10^8

Table 1. Gate and memory bit counts for SIMD, MIMD, SIMD+MIMD, and M-SIMD architectures.

Using the metric of 'gate count*time', which is a measure of the effective use of the available gate resource, the M-SIMD architecture has a higher performance over SIMD, MIMD and SIMD+MIMD arrays, Figure 15.

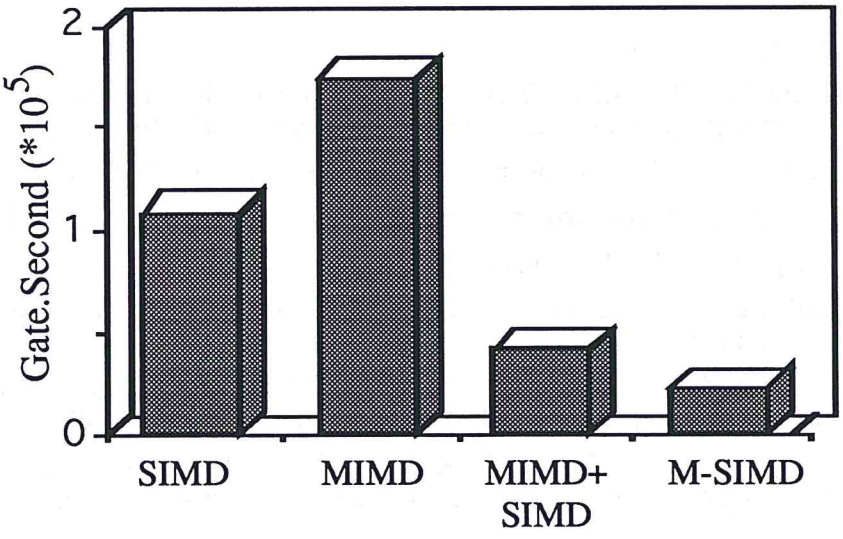

Figure 15. Gate*time on a range of architectures.

9 Concluding Remarks

We have shown how recursive estimators may be used to guide image processing operations, and how representative algorithms from a tracking task may map onto a heterogeneous massively parallel architecture. The global iconic, local iconic, and numeric operations map efficiently onto an M-SIMD computer architecture. The architecture combines two enhancements to conventional parallel architectures, heterogeneity of processor types matched to the computation at each level, and operational autonomy within an SIMD array. The cost of these enhancements in terms of hardware complexity is ~10%.

Acknowledgements

This work was supported in part by MOD (PE) under contract MGW31B/2150 administered by RARDE (IW2), the U.S. SDI ISTO under grants N00014-87-G-0241 and N000014-90-J-1919 administered by Dr K. Bromley of the Office of Naval Research, and by the ESPRIT Exploratory Action 5669, "Measure" administered by Dr. M. Coyle for DG XIII.

References

1. Hunt, D.J. The AMT DAP - a processor array in a workstation environment. Computer Systems Science and Engineering April 1989; 4 (2) : 107-114.
2. Hillis, W.D. The Connection Machine, MIT Press, Cambridge , 1985.
3. Ballard, D.H. and Brown, C.M. Computer Vision, Prentice-Hall , 1982.
4. Shapira, R. . IEEE Trans. Pami-6 1984; : 789-794.
5. Mott, D.H. . In Proc. 5th Int. Conf. on Robot Vision and Sensory Control, 1985, pp. 335-350.
6. Clippingdale, S.C. and Wilson, R.G. Least-Squares Image Estimation on a Multiresolution Pyramid. In Proceedings ICASSP89, 1989.
7. Matthies, L. and Kanade, T. Kalman filtering-based algorithms for estimating depth from image sequences. International Journal of Computer Vision 1989; 3: 209-236.
8. Harris, C.G. and Pike, J.M. 3D Positional Integration from Image Sequences. Image and Vision Computing May 1988; 6 (2) : 87-90.
9. Faugeras, O.D., Ayache, N., Lustman, F., and Giuliano, E. Depth and Motion Analysis: the machine being developed within Esprit Project 940. In Proceedings of IAPR Workshop on Computer Vision, 1988, pp. 35-44.

10. Marslin, R., Sullivan, G.D., and Baker, K.D. Kalman Filters in Constrained Model Based Tracking. In Proceedings of the British Machine Vision Conference BMVC91, Mowforth, P. (Ed), Springer-Verlag, 1991, pp. 371-374.

11. Rosenfeld, A. and Kak, A.C. Digital Picture Processing, Academic Press , 1982.

12. Haralick, R. and Shapiro, L.G. Segmentation and its place in machine vision. Scanning Microscopy Supplement 1988; 2: 39-54.

13. Markham, K.C. Comparison of Segmentation processes for object acquisition in infrared images. IEE Proceedings February 1989; 136 (F1) : 13-21.

14. Amdahl, G.M. Validity of the single processor approach to achieving large scale computing capabilities. In Proceedings AFIPS Spring Joint Computer Conference, April 1967, pp. 483-485.

15. Maresca, M., Lavin, M.A., and Li, H. Parallel Architectures for Vision. Proc. of the IEEE August 1988; 76 (8) : 970-981.

16. Atherton, T.J., Kerbyson, D.J., and Nudd, G.R. Passive Estimation to Objects from Image Sequences. In British Machine Vision Conference 1991 , Springer-Verlag, September 1991, pp. 343-346.

17. Minor, L.G. and Sklansky, J. Detection and Segmentation of blobs in infrared images. IEEE Trans. SMC 1981; 11: 194-201.

18. Atherton, T.J., Nudd, G.R., Clippingdale, S.C., Francis, N.D., Kerbyson, D.J., Packwood, R.A., and Vaudin, G.J. Detection and Segmentation of Blobs using the Warwick Multiple-SIMD Architecture. In Parallel Architectures for Image Processing, SPIE, February 1990, pp. 96-104.

19. Bar-Shalom, Y. and Fortmann, T.E. Tracking and Data Association, Vol. 179, Academic Press , Mathematics in Science and Engineering, 1988.

20. Golston, J.E., Moss, R.H., and Stoecker, W.V. Boundary detection in skin tumour images: an overall approach and a radial search algorithm. Pattern Recognition November 1990; 23 (11) : 1235-1247.

21. Sullivan, G.D., Worrall, A.D., Hockney, R.W., and , K.D.B. Active Contours in Medical Image Processing using a Networked SIMD Array Processor. In Proc. of the British Machine Vision Conference BMVC90, September 1990, pp. 395-400.

22. Nudd, G.R., Kerbyson, D.J., Atherton, T.J., Francis, N.D., Packwood, R.A., and Vaudin, G.J. A Massively Parallel Heterogeneous VLSI Architecture for MSIMD Processing. In *Algorithms and Parallel VLSI Architectures*. Elsevier North Holland, 1991.

23. Bar-Shalom, Y., Chang, K.C., and Shertukde, H.M. Performance Evaluation of a Cascaded Logic for Track Formation in Clutter. IEEE Trans. on AES November 1989; 25 (6) : 873-878.

24. Kimme, C., Ballard, D., and Sklansky, J. Finding Circles by an Array of Accumulators. Proc. ACM 1975; 18 (2) : 120-122.

25. O'Gorman, L. and Sanderson, A.C. The Wedge Filter Technique for Convex Boundary Estimation. IEEE Trans. on Pattern Analysis and Machine Intelligence May 1985; 7 (3) : 326-332.

26. Therrien, C.W. Decision, Estimation, and Classification, John Wiley, New York, 1989.