

MILITARY TECHNICAL COLLEGE
CAIRO - EGYPT



7th INTERNATIONAL CONF. ON
AEROSPACE SCIENCES &
AVIATION TECHNOLOGY

DIFFERENT APPROACHES TO MOTION PARAMETERS ESTIMATION FROM IMAGE SEQUENCES

M. K. EL-MAHY* SAYED F. BAHGAT ** A. EL NASORY***

Abstract:

This paper reviews the basic techniques of motion information recovery from image sequences. It highlights the recursive method of estimating motion parameters as they are computationally less complex than nonrecursive ones. This paper focuses on the use of an extended Kalman filter (EKF) in 2-D image processing. A straightforward extension of one-dimensional Kalman filter techniques would result in a number of state variables proportional to n for the filtering of an $n \times n$ digital image. To overcome this problem, some approximations have been used to reduce the computational load of the algorithm but at the expense of its accuracy.

This paper investigates also the parallel implementation of the EKF in 2-D image processing problem for a range of transputer topologies to enable practical realizations. The parallel implementation results are presented and the relative speedup and efficiency measures are calculated to evaluate the performance.

Key Words:

Image tracking, Kalman Filtering, Parallel processing, Transputers.

1. Review of Motion Parameters Estimation Techniques :

This paper presents some of the basic concepts used to extract motion parameters of a vehicle from sequence of images. It is assumed that image plane coordinates of feature point can be extracted and that point correspondences are available from image to image. It is also assumed that only one camera is used.

This problem has been solved by a number of researches under certain assumptions, the usual one being that translational motion parameters can be determined only within

* Phd, MIEE, CEng, Egyptian Armed Forces, Cairo, Egypt.

** Associate Professor, Military Technical College, Cairo, Egypt.

*** Post Graduate Student, Military Technical College, Cairo, Egypt.

A more applicable discrete-time nonlinear stochastic system [16] can be described by the following state-space description:

$$x_k = \varphi_{k-1}(x_{k-1}) + \gamma_{k-1}(x_{k-1})w_{k-1} \quad (1)$$

$$y_k = h_k(x_k) + v_k \quad (2)$$

where $\varphi_{k-1}(x_{k-1})$ and $h_k(x_k)$ are, in general, nonlinear functions and $\gamma_{k-1}(x_{k-1})$ is not constant. The process noise w_{k-1} and the measurement noise v_{k-1} are assumed to be zero mean, white Gaussian random sequences. The initial state x_0 is a Gaussian random vector with zero mean and covariance P_0 . It is assumed that w_k , v_k , x_k , x_0 are mutually uncorrelated and hence independent because of the Gaussian property, $E[w_k w_j^T] = Q_k \delta_{kj}$, and $E[v_k v_j^T] = R_k \delta_{kj}$, where δ_{kj} is the Kronecker delta function and E denotes the expectation operator.

To deal with these nonlinear functions, let Φ_k , H_k , and Γ_k denote the Jacobian matrices evaluated at $\hat{x}_{k/k-1}$, defined as:

$$\begin{aligned} \Phi_k &= \left. \frac{\partial \varphi_k(x_k)}{\partial x_k} \right|_{x_k = \hat{x}_{k/k-1}} \\ H_k &= \left. \frac{\partial h_k(x_k)}{\partial x_k} \right|_{x_k = \hat{x}_{k/k-1}} \\ \Gamma_k &= \gamma_k(\hat{x}_{k/k-1}) \end{aligned} \quad (3)$$

(Note: these matrices are strictly functions of $\hat{x}_{k/k-1}$ but are written thus for convenience. $\hat{x}_{k/k-1}$ denotes the conditional mean estimate of x_k based on the measurement y_{k-1}).

When the nonlinear functions in these equations are continuous, the linear dynamics and nonlinear measurement functions can then be linearized using Taylor series expansion about the predicted state estimate. Hence, the EKF equations are given by:

Prediction Mode.

$$\begin{aligned} \hat{x}_{k/k-1} &= \Phi_{k-1} \hat{x}_{k-1/k-1} \\ P_{k/k-1} &= \Phi_{k-1} P_{k-1/k-1} \Phi_{k-1}^T + \Gamma_{k-1} Q_{k-1} \Gamma_{k-1}^T \end{aligned} \quad (4)$$

Innovation Sequence.

$$v_k = y_k - \hat{y}_{k/k-1} = y_k - H_k \hat{x}_{k/k-1} \quad (5)$$

The measurement Jacobian matrix is given by:

$$H_k(\hat{x}_{k/k-1}) = \left. \frac{\partial h_k(x_k)}{\partial x_k} \right|_{x_k = \hat{x}_{k/k-1}} \quad (6)$$

The covariance of the innovation sequence is given by:

$$S_k = H_k P_{k/k-1} H_k^T + R_k \quad (7)$$

Kalman Gain.

$$K_k = P_{k/k-1} H_k^T S_k^{-1} \quad (8)$$

Correction Mode.

$$\begin{aligned} \hat{x}_{k/k} &= \hat{x}_{k/k-1} + K_k \nu_k \\ P_{k/k} &= [I - K_k H_k] P_{k/k-1} \end{aligned} \tag{9}$$

with initial conditions: $\hat{x}_{0/0} = E[x_0]$ and its initial covariance $P_{0/0} = Var[x_0]$.

It is clear from equation 9 and equation 8 that the structure of the covariance and the gain computation equations in the EKF are identical to those of the structure of linear KF where the functions $\varphi_k(x_k)$, $\gamma_k(x_k)$, and $h_k(x_k)$ replaced by their Jacobians (derivatives) of the nonlinear functions. The approximate conditional mean signifies that the updated state is computed as a linear combination of the prediction and the innovation. It is important to note that the EKF depends on the current state estimate and thus cannot be computed off-line. Furthermore, the estimation accuracy achieved is trajectory dependent because the error covariance matrices are functions of the time-history of \hat{x}_k . Note also that the EKF is a sub-optimal filter as $P_{k/k}$ is an approximate mean-square error rather than a covariance, as $\hat{x}_{k/k}$ is not the exact conditional mean [17]. For best results when the EKF filter is used, the covariance matrix $P_{k/k}$ should be initially small and remains small relative to the dynamical noise Q_k or the measurement noise R_k [18].

A major advantage of Kalman filtering is that the associated error covariance matrices can be used for the purposes of providing gating (measurement correlation) information. Also the gain sequence is based on the assumed motion and measurement noise models, which can be varied for various motion and measurement regimes i.e. adaptive filtering. It is apparent that the Kalman filter algorithm consists of a time update (the prediction stage) followed by a measurement update (the correction stage). For the linear case these stages are performed in a manner that minimizes the mean square error of the state estimate. In the nonlinear case, however, this optimality is lost because of the various approximations. Following is a description of the linear Kalman filter, after which the nonlinear case is presented.

III. Motion, Image and Noise Models

The models adopted in this paper are based upon a body-centered coordinate system defined in reference [14]. We are interested in recovering the body rates (i.e. u, v, w, p, q, r), instead of the 3D structure of the image object and its object based rates, as in [13]. The coordinate system (O, x, y, z) is defined as in figure 1 and is assumed to be rigidly attached to the imaging system at its focal point O .

III.1. Motion Model

The motion of a rigid object can be resolved into two components, a translation and a rotation. The translation is denoted by $T = (u, v, w)^T$ and the angular velocity by $\omega = (p, q, r)^T$. If L is the vector $(x, y, z)^T$ of a point, then that point will have a velocity given by

$$V = -T - \omega \times R \tag{10}$$

Then, the rectangular velocity components can be written as:

$$\dot{x} = -u - qz + ry \tag{11}$$

a scale factor. Other limitations include that the environment is modeled as a flat plane, which may be suitable for a highly flying air vehicle, but is inappropriate for land vehicles.

Two approaches are available for the measurement of motion parameters from image sequences: discrete approach and continuous approach. In the discrete approach, only a few points are used to find the motion but it is dependent upon reliable matching of the feature points in successive images, so that an image plane trajectory is available. This method is used by several authors [1, 2, 3], where some demonstrate the minimum number of points required for a unique solution.

The continuous approach uses the whole optical flow field, which is derived from gray level gradients. This data is then used to implement a least squares algorithm based on some norm [4, 5], which leads to a set of nonlinear equations. This method is computationally expensive in the image processing phase. However, it is considered to be robust in the presence of noise due to the large amount of data used. Since the optical flow field is practically arrived at differencing image correspondences, it is effectively a discrete method.

How much noise each feature exhibits is usually taken to be some function of its image size, but is also dependent upon the actual object size and how the measurements are modeled. Another consideration is whether Gaussian noise model is valid when used for an imaging system. In different situations certain degradation results; if the scene is moving rapidly, image blurring occurs, while if the scene moves slowly, quantization effects dominate. As such, no author has quantified the effects of noise on an imaging system, so it seems that this model is good enough for the time being.

For the first approach, the correspondence problem is not an easy one, but once it has been overcome, the parameter estimates tend to be of higher accuracy. The second method is better suited for images containing multiple moving objects, because of the use of segmentation. Indeed, it seems that optical flow information is essential for scene segmentation [6]. A hybrid of these two methods is proposed in [7], where a test for rigidity of the image points is included; however, when noise is present the method is not very robust.

The solution methods so far encountered, define object motion as the rigid body transformation that takes an object from one spatial position in one image to its position in the next image. This transformation is a translation followed by a rotation, or vice-versa. An approximate values for the translation and rotation velocities are available as time can be measured. These values are less accurate if a limited number of images is used, as they can be affected by noise.

In [3, 8, 9, 10], Tsai and Huang devise a method for the estimation of motion parameters from two images. They manipulate the retinal motion equations, for the discrete case, to derive eight "essential parameters", which can be solved with eight linear equations. This is dependent upon eight image correspondences being available. More points can be included for which a least squares solution is employed. Most importantly, they show that the eight essential parameters are unique depending upon the spatial arrangement of the points.

Fang and Huang [11] use a similar approach using a nonlinear method where six point correspondences are required. They show how to extract feature points and establish correspondences, but only if the motion between successive images is neither large nor small. However, the methods are still sensitive to noise.

Roach and Aggerwal [12] define the 3-D motion as product of 4×4 translation and

rotation matrices. They derive 20 nonlinear equations which relate the image coordinate, spatial coordinate and camera position, again assuming that five feature point correspondence can be established. Numerical techniques for the solution of these equations are described and a method for choosing good initial conditions is demonstrated.

An optical flow implementation is used in [6]. It is evaluated for very large number of points in the image, but this does not seem to negate the effects of noisy data. So it is suggested that a sequence of flow fields can be used.

A recursive method is employed in [13], with translation represented by a truncated Taylor series and rotation by quaternions. It demonstrates the use of a large sequence data to filter and smooth the estimates using a Kalman filter and shows that an arbitrary number of points can be used. This has several advantages as occlusion can be dealt with. A prediction of point locations on the image plane can be made, to help process the next image and determine correspondences. As a Kalman filter implementation is employed, a significant amount of noise can also be handled.

At this stage, it should be noted that all of the above methods only calculate the motion parameters to within a scale factor, due to the depth ambiguity that arises. Also, none of the above techniques employ any knowledge of the dynamics of the motion, which is certainly a critical factor when attempting to estimate a vehicle motion.

Ahmed [14] introduced a new method of extracting the motion parameters of a vehicle, with known dynamics, from a sequence of images. The images provide information about the scene from which various features, such as edges, corners and simple objects, can be identified. These features can then be utilized to form estimates of the three dimensional motion of the vehicle, under the assumption that image plane coordinates of feature points can be extracted and that point correspondences are available from image to image. Ahmed [14] used a recursive algorithm to estimate and track the motion parameters.

The motivation behind using the recursive algorithm was that this approach requires much less computation when additional data is introduced, for instance as each frame is processed. Also, these estimates can easily be extrapolated forwards in time as an aid to processing subsequent images. Another consideration is that, in this case, approximate filters must be used because of the nonlinear character of the plant and measurement processes. If these processes are modeled only approximately, then the limited memory aspects of such filters can be advantageous, as small deviations from the process models are absorbed by the noise model. The limited memory of such filters also means that all the available data is utilized but is forgotten in an exponential manner, so that more recent measurements have a greater effect on the current estimate.

II. Review of Extended Kalman Filter Algorithm

The extended Kalman filter (EKF) is a standard nonlinear sub-optimal filtering algorithm which is also used to recursively estimate the state vector using the dynamics and the measurement models [15]. It models the unknown parameters as additional state variables to be estimated and the convergence of the filter is assessed by heuristic methods.

Linearization of the nonlinear functions is essential to obtain a real time implementable algorithm. The EKF is constructed by linearizing either the system equation and/or the measurement one about the predicted state estimate using the first two terms of a Taylor series expansion. This produces a constant term and a first derivative term, known as the Jacobian matrix.

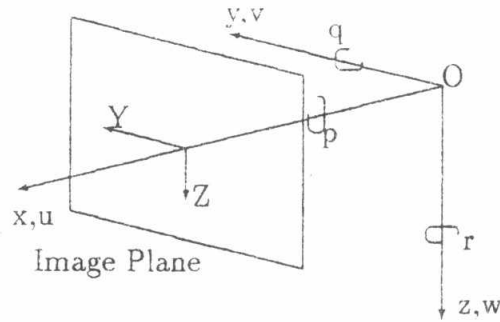


Figure 1: Axis System.

$$\dot{y} = -v - rx + pz \tag{12}$$

$$\dot{z} = -w - py + qx \tag{13}$$

The vehicle to which the camera is attached, has certain motion characteristics that are defined by a set of linear, or nonlinear differential equations. These equations help form a model of the camera motion. Such model should be involved in the motion parameter estimation.

III.2. Image Model

The imaging model that is used in most research is the central projection model as shown in figure 1. The image plane coordinates of a point are defined by:

$$Y = f \frac{y}{x} + n_Y \tag{14}$$

$$Z = f \frac{z}{x} + n_Z \tag{15}$$

where f , n_Y and n_Z are defined respectively as the camera focal length (for example $f = 1$) and white Gaussian noise quantities. Differentiating equations 14 and 15 with respect to time and ignoring the noise terms, gives the result:

$$\dot{Y} = \frac{\dot{y}}{x} - \frac{y\dot{x}}{x^2}, \dot{Z} = \frac{\dot{z}}{x} - \frac{z\dot{x}}{x^2} \tag{16}$$

Substituting equations 11 to 13 into equation 16 yields:

$$\dot{Y} = \frac{1}{x}(-v + uY) + pZ + qYZ - r(1 + Y^2) \tag{17}$$

and

$$\dot{Z} = \frac{1}{x}(-w + uZ) - pY + q(1 + Z^2) - rYZ \tag{18}$$

It is concluded from the equations that the optical flow is dependent upon the six motion parameters and the structure of imaged environment. But, only five parameters can be determined, because of the depth ambiguity that arises. This ambiguity comes about because any constant multiple of the spatial coordinates gives the same image. However, this only effects the translation parameters.

For example, a point at $(2x, 2y, 2z)$ with translation rates of $(2u, 2v, 2w)$ will produce the same image plane coordinates and velocities as a point at (x, y, z) moving at (u, v, w) .

III.3. Noise Model

It is assumed that the measured image coordinates are corrupted with white zero mean Gaussian noise. Thus, it is assumed that n_Y and n_Z are uncorrelated and identically distributed with variance σ^2 . The joint probability density function is given by:

$$f(Y, Z) = \frac{1}{2\pi\sigma^2} \lambda \quad (19)$$

where λ is given by:

$$\lambda = \exp\left(-\frac{(Y - \mu_y)^2 + (Z - \mu_z)^2}{2\sigma^2}\right) \quad (20)$$

IV. Motion Parameters Estimation from 2-D Image Sequences Using EKF Algorithm

It is assumed that a camera moving like in [14] in a rigid static space, and the parameters of interest to us are the actual body rates u, v, w, p, q, r , instead of the inertial rates determined in [13]. Now, it is not possible to determine all the motion parameters because of the depth ambiguity associated with recovering 3-D motion from a 2-D scene. This results in an unknown scale factor that cannot be determined without prior knowledge of any one of the translation rates or structure parameters. To overcome this problem, one of translation rates or structure parameters is used to normalize remaining translation parameters.

Consider for example the motion model of a simple land vehicle that has a forward speed (u) of 50ms^{-1} and $\dot{u} = 0$. In this example, the normalizing parameter should be the forward velocity u . The parameters v/u and w/v can respectively be regarded as approximation. Then, the state space equations that describes this model are given by:

$$\begin{pmatrix} \dot{v}/u \\ \dot{r} \end{pmatrix} = \begin{pmatrix} -3.0192 & 0.0059 \\ 5.5830 & -2.9227 \end{pmatrix} \begin{pmatrix} v/u \\ r \end{pmatrix} \quad (21)$$

As the motion is constrained to the horizontal plane, no rotation is considered (i.e. $w = p = q = 0$). Also the higher derivative terms of v/u and r are not considered. Thus:

$$\left(\frac{\dot{v}}{u}\right) = -3.0192 \left(\frac{v}{u}\right) + 0.0059r \quad (22)$$

$$\dot{r} = 5.5830 \left(\frac{v}{u}\right) - 2.9227r \quad (23)$$

$$\frac{\dot{x}_i}{u} = -1 + rY_i \frac{x_i}{u} \quad (24)$$

$$\dot{Y}_i = \frac{u}{x_i} \left(\frac{v}{u} + Y_i\right) - r(1 + Y_i^2) \quad (25)$$

$$\dot{Z}_i = \frac{u}{x_i} Z_i - rY_i Z_i \quad (26)$$

The state transition matrix, Φ_k is given by:

$$\phi_k = \begin{pmatrix} \varphi_0 & 0 & \cdots & 0 \\ \varphi_{1_1} & \varphi_{2_1} & & \vdots \\ \vdots & & \ddots & \vdots \\ \varphi_{1_m} & \cdots & \cdots & \varphi_{2_m} \end{pmatrix} \quad (27)$$

where

$$\varphi_0 = \begin{pmatrix} -3.0192 & 0.0059 \\ 5.5830 & -2.9227 \end{pmatrix} \quad (28)$$

$$\varphi_{1_i} = \begin{pmatrix} 0 & Y_i(x_i/u) \\ u/x_i & -(1 + Y_i^2) \\ 0 & -Y_i Z_i \end{pmatrix} \quad (29)$$

$$\varphi_{2_i} = \begin{pmatrix} Y_i r & (x_i/u)r & 0 \\ \varphi_{21} & u/x_i - 2Y_i r & 0 \\ -Z_i/(x_i/u)^2 & -Z_i r & \varphi_{33} \end{pmatrix} \quad (30)$$

where $\varphi_{21} = (v/u - Y_i)/(x_i/u)^2$ and $\varphi_{33} = u/x_i - Y_i r$.

A solution can now be obtained by implementing an extended Kalman filter. The initial state vector x_0 and its covariance matrix are chosen as:

$$\hat{x}_0 = \begin{pmatrix} 0 \\ 0 \\ 50 \\ 0 \\ 0 \\ 50 \\ 0 \\ 0 \end{pmatrix} \quad (31)$$

$$P_0 = \begin{pmatrix} 10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 10 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 10 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 10 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 10 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 10 \end{pmatrix} \quad (32)$$

i.e. $P_0 = 10I$ where I is the identity matrix.

The non-zero elements biasing in the initial state estimate are required to prevent singularities arising during the construction of $\varphi(x_0)$. The image plane noise variance R_k is chosen to be 10^{-6} .

Figure 2 shows the effects of dynamical noise variance Q_k upon the estimation parameters. For $Q_k = 10^{-4}$, it is evident that the sideslip angle and yaw rate are not accurately estimated. Additionally, the estimates of the image plane coordinates (particularly Z_i) exhibit a great deal of sensitivity. These effects are primarily caused by Q_k being the

dominant term in the predicted error covariance equation, equation 4, that results in a large gain and thus the sensitivity to any error in the measurements.

With $Q_k = 10^{-6}$ the sensitivity is considerably reduced, as are the errors in the estimation of v/u and r . Also note that the estimate of x_i/u seems to be converging more rapidly towards their true values. Reducing Q_k to 10^{-8} has no obvious effect on estimation accuracy, except that x_i/u converges much more rapidly. Further reduction in Q_k yields no significant improvement in accuracy. This is thought to be due to R_k being the dominant term in equation 7 as Q_k is reduced in size. This also explains the initial transients of v/u for the smaller values of Q_k .

Out of curiosity, it was decided to vary the value for the measurement noise covariance, R_k , while holding $Q_k = 10^{-10}$, for which the results are shown in figure 3. For $R_k = 10^{-4}$, all the quantities are estimated well, except the ranges (x_i/u). These converge slowly towards their true values because the reduction in the error covariance is slowed by large values of R_k . This manifests itself in the estimation of the image plane coordinates which tend to diverge at the 20th iteration and only begin converging again when the range estimates have over shot their true values.

For $R_k = 10^{-6}$, all the motion and image plane estimates seem to converge quickly towards their true values, while the range estimate converges after the 17th iteration. Reducing R_k to 10^{-8} indicates no significant effect upon the estimate convergence, except that the range estimate converges somewhat quickly. Smaller values of R_k cause a faster reduction in the error covariances, which improves the convergence rate of the range estimates. It was found that further reduction in R_k resulted in bad conditioning of the matrix inversion in equation 8.

It is noted that using values for R_k that are inconsistent with the actual measurement noise statistics, produces error covariances that cannot be used for purposes of track assignment and correlation.

V. Parallel Implementation of EKF Algorithm

The noisy data for the center of the target for each image is available to be used to evaluate the parallel implementation algorithm of the extended Kalman filter EKF in the planer image.

V.1. Target Hardware

Transputer target systems can be constructed from hardware building blocks called TRAMs [19]. TRAM is a self contained unit which uses the transputer serial links to interface to the outside world. TRAM modules plug directly into TRAM motherboards such as PC/AT or VME bus format cards.

TRAMs can be either compute TRAMs (combining a transputer and memory) or application specific TRAMs (interfacing a transputer to application hardware), e.g. graphics, image processing, Ethernet, or SCSI.

The target hardware used in this work is a Kane Computing HEPC1 (PC TRAM Motherboard) which is a 10 slot motherboard compatible with IBM XT or AT type PCs. The Inmos links are configured by jumpers which are set by default in the usual 2-1 pipe with Slot 0 - Link 0 booting from the host via C012 link adaptor. A 37-way D-type connector is fitted to cater for larger networks. The TRAM slots are connected into a

pipeline using two of the four links from each slot. The remaining two links from each processor are connected to the C004 which allows link connections to be softwired together to form other transputer topologies. The processors are T800-25MHz with 2Mb of RAM.

V.2. Performance Metrics

Using more processors significantly reduce the execution time. Speedup (S_p) is defined as the ratio of the single processor execution time T_1 for a parallel algorithm to the execution time for the same algorithm on p processors T_p , i.e.

$$S_p = T_1/T_p \quad (33)$$

therefore, the best speedup is the one close to be linear in p , i.e. $S_p = p$.

The efficiency E_p relates the speedup to the number of processors as a percentage, i.e.

$$E_p = (S_p/p) \times 100\% \quad (34)$$

If the efficiency is close to unity (the ideal speedup is 100%), it indicates that the hardware effectively utilizes all processors, while low values of efficiency mean that all or a number of processors are idle for significant amounts of time [20, 21, 22].

V.3. Software

The applications in this paper are programmed in *Parallel C* [23] which is a full implementation of ANSI C, with library functions which make available the concurrency and communications features of the transputer. It is worth noting that OCCAM does not make any implicit assumption about the sequential nature of the flow through a program as does *Parallel C*. Users must declare explicitly any sequentialism when they decide to use in OCCAM. Also, it is worth noting that the latest generation OCCAM compiler is itself written in C.

The fundamental unit in OCCAM is the process while in *Parallel C* is the task. As with other languages, the application is declared as a set of tasks which are independently compiled and linked. The set of tasks can then be mapped onto a processor network, with tasks communicating via channels, which can be realized as transputer links for tasks on adjacent processors, or via RAM for tasks which are sharing a processor. The user has to write a configuration file which describes the system to be built by listing all the physical processors in the system, the wires connecting them, the tasks to be loaded into the system and their logical interconnections. This file is used to drive the configurer program, which is supplied with the *Parallel C* compiler, to produce executable files for complicated system configurations, containing many tasks and many transputers. The output from the configurer can be run directly using the *afserver*.

A logical (heuristic) mapping strategy is used to schedule the subtasks on the transputers because the parallelism is explicit in the algorithm. The association of a group of tasks with a processor is fixed and it is called a static allocation [20]. The parallel algorithm that is most effective is difficult to select and so different options have to be tried until the lowest execution time is obtained. This includes task scheduling, balancing loads, minimizing communications among the processors, blocking transferred data, and avoiding duplicated tasks.

Execution Time		Performance	
single	network	S_p	E_p
473 μ s	283 μ s	1.67	83%

Table 1: Performance of 2 Transputers.

Execution Time		Performance	
single	network	S_p	E_p
482 μ s	250 μ s	1.93	64%

Table 2: Performance of 3 Transputers.

The parallel EKF algorithm was applied to the example of section IV. Simulations were initially mapped onto single transputer using the HEPC1. The execution time for the EKF running sequentially on a single transputer is 422 μ s. One of the useful features of the transputer is that tasks can be allocated to separate processors, or may share a processor, without any recompilation of the task code. Hence a task network can be implemented, debugged, and tested on a single processor, and then distributed to a multi-processor environment once complete [20].

The execution times for all algorithms running on two to four-transputer networks shown in Tables 1 to 3. For the two transputer pipeline, the absolute speedup is 1.67 with an absolute efficiency of 83% (i.e. the two transputers have produced a 83% linear speedup). When three transputers are used, the absolute speedup is 1.93 with an absolute efficiency of 64%. Using four processors, the speedup is 2.44 with an absolute efficiency of 60%.

VI. Conclusions

This paper has briefly reviewed the basic concepts of motion information recovery from a sequence of images. It highlights the method of estimating motion parameters that manage to incorporate a recursive algorithm. This has the advantages that all previous information about the motion gleaned from previous images can aid in estimating the current motion without any substantial increase in the complexity of the motion estimation procedure. The method of tracking the image points would enable the image

Execution Time		Performance	
single	network	S_p	E_p
559 μ s	229 μ s	2.44	60%

Table 3: Performance of 4 Transputers.

processing algorithm to reduce its search area and be executable at a higher speed. It is desired to have the whole algorithm work in real time. However, it has been shown that the tracking problem causes a substantial increase in processing time when a significant number of points are tracked.

The processing time of the algorithm can be reduced, if for instance the motion model is simplified. This may, however, affect the robustness of the estimation as less information is available. At the moment the algorithm requires at least eight points to work properly, it may become unpredictable if points are lost due to occlusion, or the environment not being very rich in terms of image tokens.

Therefore, this paper has also investigated parallel implementations for the EKF in 2-D image processing on transputer arrays; full-linking transputer topologies with $N(N-1)/2$ links for $N = 1 : 4$, partially-linking transputer topologies and multi-transputer arrays, assuming that the noisy data for the center of the target for each image is available. Although execution time is significantly reduced by increasing the number of processors, the study illustrates that more reduction can be obtained by more balancing computations. The nonlinearities that are represented by the Jacobian matrix were computed on the same processor as the state vector, thereby significantly reducing the communication overhead.

References

- [1] H C Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293, 1981.
- [2] A Z Meiri. On monocular perception of 3-d moving objects. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, PAMI-2, Nov. 1980.
- [3] R Y Tsai and T S Huang. Uniqueness and estimation of three dimensional motion parameters of rigid objects with curved surfaces. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, PAMI-6, Jan. 1984.
- [4] A R. Bruss and B K P Horn. Passive navigation. *Computer Vision, Graphics, and Image Processing*, 1983.
- [5] K Prazdny. Determining the instantaneous direction of motion from optical flow generated by a curvilinear moving observer. *Computer Graphics and Image Processing*, 17, 1981.
- [6] G Adiv. Determining Three-Dimensional. Motion and structure from optical flow generated by several moving objects. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, PAMI-7, July 1985.
- [7] A Mitiche. On kineopsis and computation of structure and motion. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, PAMI-8, Jan. 1986.
- [8] R Y Tsai and T S Huang. Estimating 3-d motion parameters of a rigid planar patch. *IEEE Trans. on Acoustics, Speech and Signal Processing*, ASSP-29, Dec. 1981.

- [9] T S Huang R Y Tsai and W L Zhu. Estimating 3-d motion parameters of a rigid planar patch, 2: Singular value decomposition. *IEEE Trans. on Acoustics, Speech and Signal Processing*, ASSP-30, Aug. 1982.
- [10] R Y Tsai and T S Huang. Estimating 3-d motion parameters of a rigid planar patch, 3: Finite point correspondences and the three view problem. *IEEE Trans. on Acoustics, Speech and Signal Processing*, ASSP-32, April 1984.
- [11] J Fang and T S Huang. Some experiments on estimating the 3-d motion parameters of a rigid body from two consecutive image frames. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, PAMI-6, Sept. 1984.
- [12] J Roach and J Aggerwal. Determining the movement of objects from a sequence of images. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, PAMI-2, Nov. 1980.
- [13] S Chandrashekhara T J Broida and R Chellappa. Recursive 3-d motion estimation from a monocular image sequence. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, 1990.
- [14] M Ahmed. M.Phil/PhD Transfer Report. Technical report, Control and Guidance Systems Group, Royal Military College of Science, Cranfield University, Shrivenham, Swindon, UK, April 1992.
- [15] Maybeck P S. *Stochastic Models, Estimation and Control*, volume 1. Academic Press, 1979.
- [16] Andrew H Jazwinski. *Stochastic Processes and Filtering Theory*. Academic Press, 1970.
- [17] Bar-Shalom Y and Fortmann T E. *Tracking and Data Association*, volume 179 of *Mathematics in Science and Engineering*. Academic Press, Inc., USA, 1988.
- [18] Robins A J. The Extended Kalman Filter - And its use in estimating aerodynamic derivatives. *Aerospace Dynamics*, (9):16-24, January 1984.
- [19] El-Mahy. Parallel Processing Techniques Using Transputers. *Air Defence Magazine*, 68:50-56, February 1996.
- [20] Nocetti D F G. *Parallel Processing in Digital Flight Control*. PhD thesis, University of Wales, Bangor, UK, February 1991.
- [21] Eager D L; Zahorjan J; and Lazowska E D. Speedup Versus Efficiency in Parallel Systems. *IEEE Transactions on Computers*, C-38(3):408-422, March 1989.
- [22] Maguire L P; Irwin G W; and Lightbody G. Mapping Control Algorithms onto Transputer Arrays. In *IEE Colloquium on 'Transputer Applications', Digest NO. 129*, pages 5/1-5/10, 1989.
- [23] Scotland 3L Ltd, editor. *Parallel C User Guide*. 3L Ltd, Scotland, November 3 1989.

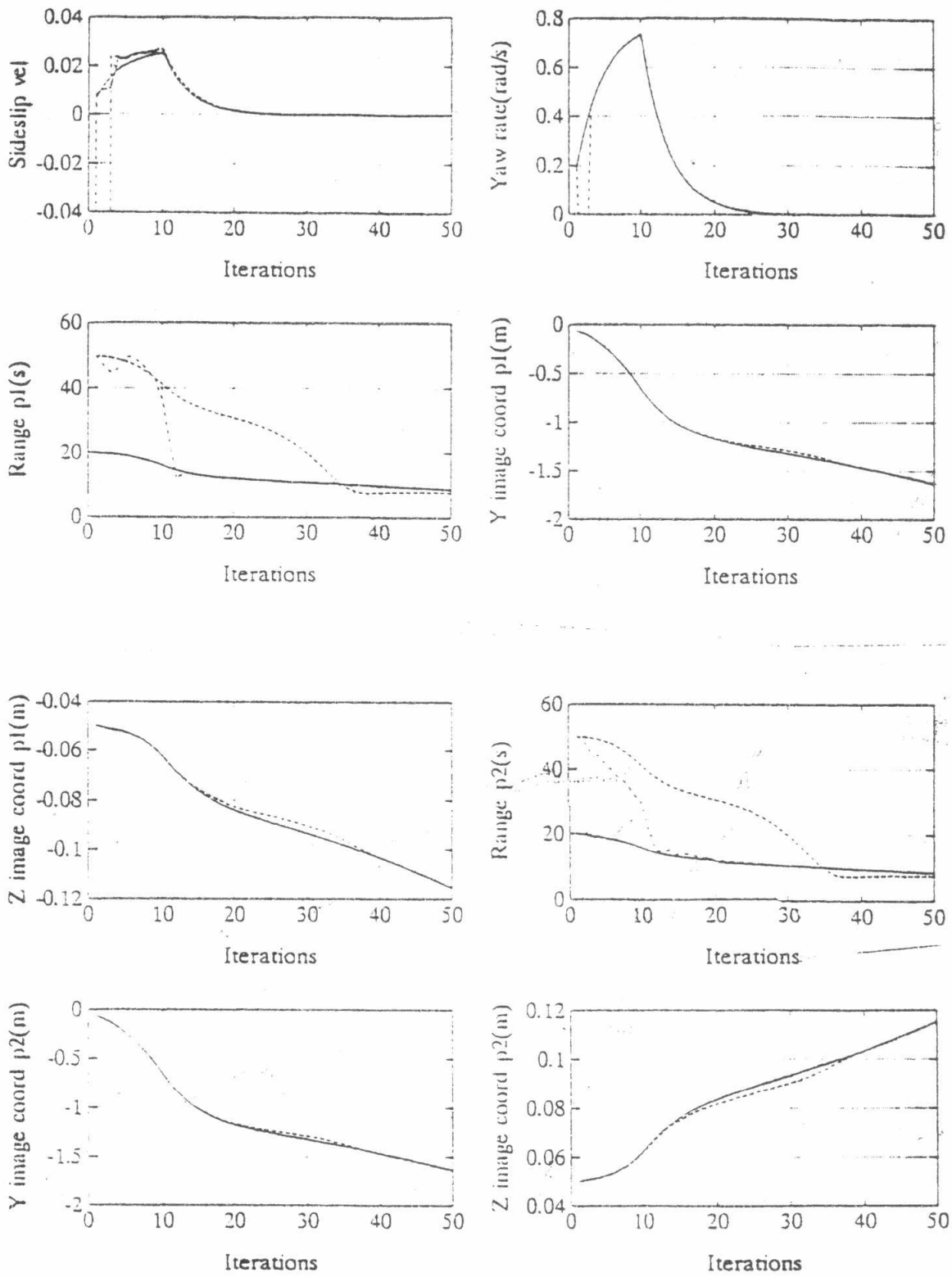


Figure 3: Effect of varying R (10^{-4} , 10^{-6} , 10^{-8}).

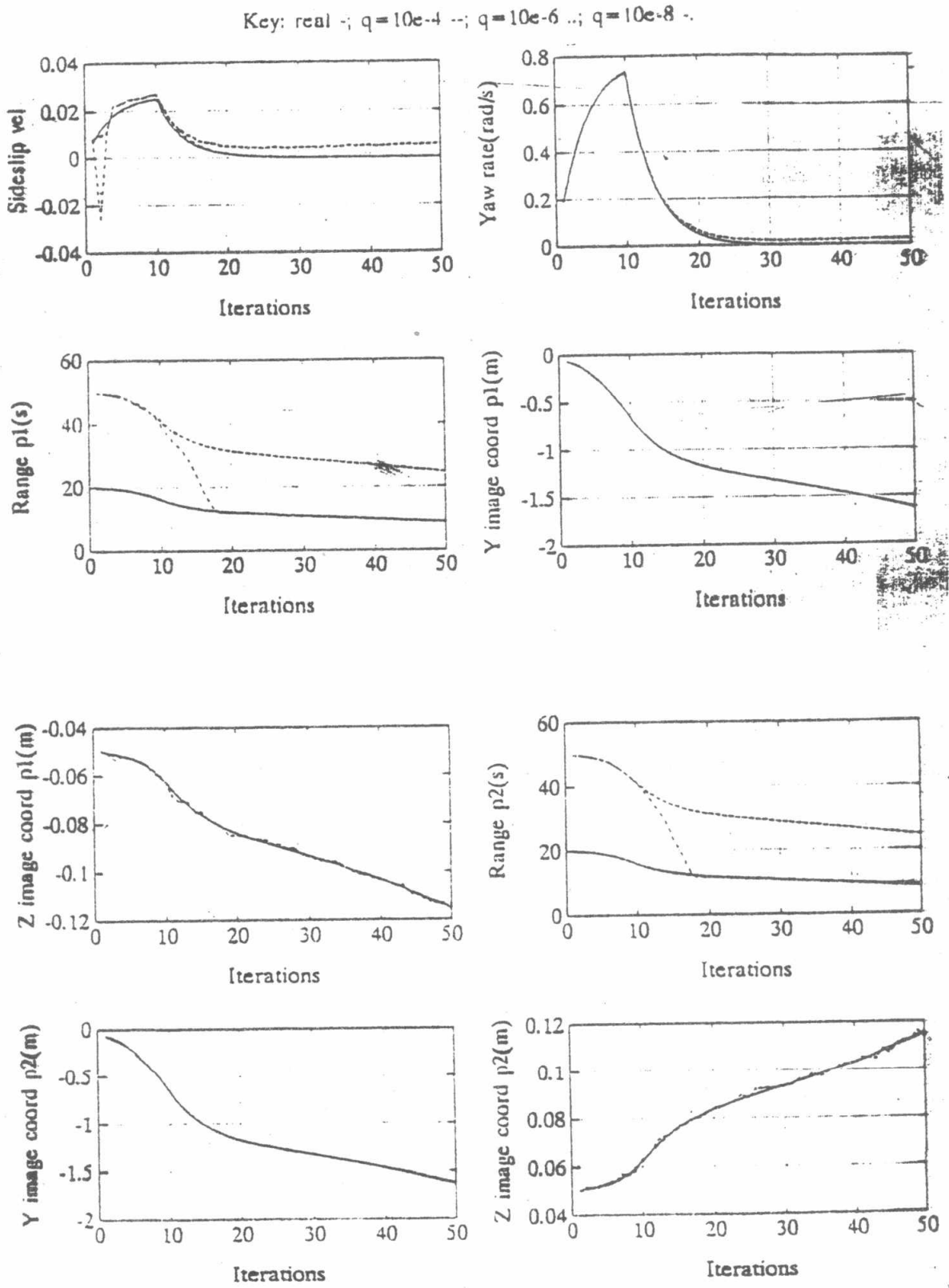


Figure 2: Effect of varying Q (10^{-4} , 10^{-6} , 10^{-8}).