International Journal of Computer Vision 28(3), 261–277 (1998)

Estimating the Focus of Expansion in Analog VLSI

IGNACIO S. MCQUIRK*
Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02139
ig@rle-vlsi.mit.edu

BERTHOLD K.P. HORN
Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA 02139
bkph@ai.mit.edu

HAE-SEUNG LEE
Microsystems Technology Laboratories, Massachusetts Institute of Technology, Cambridge, MA 02139
hslee@mtl.mit.edu

JOHN L. WYATT, JR.
Research Laboratory of Electronics, Massachusetts Institute of Technology, Cambridge, MA 02139
wyatt@rle-vlsi.mit.edu

Received October 9, 1996; Revised September 8, 1997; Accepted September 9, 1997

Abstract. In the course of designing an integrated system for locating the focus of expansion (FOE) from a sequence of images taken while a camera is translating, a variety of direct motion vision algorithms based on image brightness gradients have been studied (McQuirk, 1991, 1996b). The location of the FOE is the intersection of the translation vector of the camera with the image plane, and hence gives the direction of camera motion. This paper describes two approaches that appeared promising for analog very large scale integrated (VLSI) circuit implementation. In particular, two algorithms based on these approaches are compared with respect to bias, robustness to noise, and suitability for realization in analog VLSI. From these results, one algorithm was chosen for implementation. This paper also briefly discuss the real-time analog CMOS/CCD VLSI architecture realized in the FOE chip.

Keywords: focus of expansion, motion vision, passive navigation, analog VLSI

1. Introduction

In recent years, some attention has been given to the potential use of custom analog VLSI chips for early vision processing problems such as optical flow (Tanner and Mead, 1986), smoothing and segmentation (Yang and Chiang, 1990; Keast and Sodini, 1993) orientation (Standley, 1991), depth from stereo (Hakkarainen and Lee, 1993), edge detection (Dron, 1993) and alignment (Umminger and Sodini, 1995). The key features of early vision tasks such as these are that they involve performing simple, low-accuracy operations at each pixel in an image or pair of images, typically resulting in a low-level description of a scene useful for higher level vision. This type of processing is often well suited to implementation in analog VLSI, resulting in compact,

Funding for this work was provided by the National Science Foundation and DARPA under contracts MIP-8814612 and MIP-9117724. Ignacio S. McQuirk was supported by an NSF graduate fellowship and a Cooperative Research Fellowship from AT&T Bell Labs.

*Present address: Maxim Integrated Products, Sunnyvale, CA 94086.

high speed, and low power solutions. Through a close coupling of processing circuitry with image sensors, these chips can exploit the inherent parallelism often exhibited by early vision algorithms, allowing for an efficient match between form and function. This paper details some of the algorithms as well as the analog VLSI implementation developed in the application of this focal-plane processing approach to the early vision task of passive navigation.

An important goal of motion vision is to estimate the 3D motion of a camera based only on the measured time-varying images. Traditionally, there have been two basic approaches to this problem. In feature-based methods, an estimate of motion and scene structure is found by establishing the correspondence of prominent features such as edges, lines, etc., in an image sequence (Jain, 1983; Dron, 1993). In motion field-based methods, the optical flow (Horn and Schunk, 1981) is used to approximate the projection of the three-dimensional motion vectors onto the image plane and from this an estimate of camera motion and scene depth can be found (Bruss and Horn, 1983). Both the optical flow calculation and the correspondence problem have proven to be difficult in terms of reliability and, more importantly, implementation. In keeping with our paradigm of local, low-level, parallel computation, we have explored methods which directly utilize image brightness information to recover motion (Horn, 1990; McQuirk, 1991).

The introduction of the focus of expansion (FOE) for the case of pure translation simplifies the general motion problem substantially. The FOE is the intersection of the translation vector of the camera with the image plane. This is the image point towards which the camera is moving, as shown pictorially in Fig. 1. With a positive component of velocity along the optic axis, image features will appear to move away from the FOE and expand, with those closer to the FOE moving slowly and those further away moving more rapidly. Through knowledge of the camera parameters, the FOE gives the direction of 3D camera translation. Once the location of the FOE has been ascertained, we can estimate distances to points in the scene being imaged. While there is an ambiguity in scale, it is possible to calculate the ratio of distance to speed. This allows one to determine the time-to-impact between the camera and objects in the scene. Applications for such a device include the control of moving vehicles, systems warning of imminent collision, obstacle avoidance in mobile robotics, and aids for the blind.

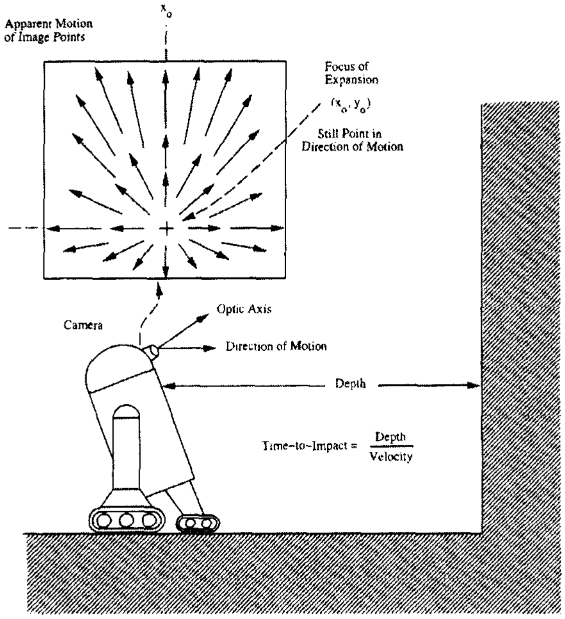
Apparent Motion of Image Points
x_o
Focus of Expansion
(x_o, y_o)
Still Point in Direction of Motion
y_o
Optic Axis
Camera
Direction of Motion
Depth
Time-to-Impact = $\frac{\text{Depth}}{\text{Velocity}}$

Figure 1. Illustration of the passive navigation scenario, showing the definition of the focus of expansion as the intersection in the camera frame of reference of the camera velocity vector with the image plane.

A variety of direct methods for estimating the FOE were explored for implementation in analog VLSI; two of the more promising algorithms considered are presented in this paper. We chose one for actual realization in an integrated system and the architecture used for this FOE chip will also be described.

2. The Brightness-Change Constraint Equation

The brightness-change constraint equation (BCCE) forms the foundation of various algorithms for rigid body motion vision (Negahdaripour and Horn, 1987a; Horn and Weldon, 1988) and is also the basis for the variants that we have explored for potential implementation in analog VLSI. This equation relates the observed brightness gradients in the image with the motion of the camera and the depth map of the scene. It is derived from the three basic assumptions:

- A pin-hole model of image formation.
- Rigid body motion in a fixed environment.
- Instantaneously constant scene brightness.

We proceed by formalizing each assumption in turn.

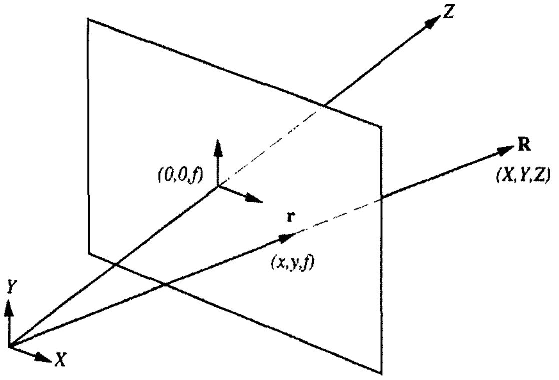

Figure 2. Viewer-centered coordinate system and perspective projection.

A Pin-Hole Model of Image Formation

Following (Bruss and Horn, 1983; Negahdaripour and Horn, 1987a; Horn and Weldon, 1988), a viewer-based coordinate system with a pin-hole model of image formation is adopted as depicted in Fig. 2. A world point

$$\mathbf{R} \equiv (X, Y, Z)^T \tag{1}$$

is mapped to an image point

$$\mathbf{r} \equiv (x, y, f)^T \tag{2}$$

using a ray passing through the center of projection placed at the origin of the coordinate system. The image plane $Z = f$, where f is the principal distance, is positioned in front of the center of projection for convenience. The optic axis is the perpendicular from the center of projection to the image plane and is parallel to the Z-axis. The x- and y-axes of the image plane are also parallel to the X- and Y-axes and emanate from the principal point $(0, 0, f)$ in the image plane.

The world point $\mathbf{R}$ and the image point $\mathbf{r}$ are geometrically related by the perspective projection equation (Horn, 1986):

$$\frac{\mathbf{r}}{f} = \frac{\mathbf{R}}{\mathbf{R} \cdot \hat{\mathbf{z}}} \tag{3}$$

Rigid Body Motion in a Fixed Environment

Assuming that the camera moves relative to a fixed environment with translational velocity $\mathbf{t} = (t_x, t_y, t_z)^T$ and rotational velocity $\omega = (\omega_x, \omega_y, \omega_z)^T$, the motion of a world point $\mathbf{R}$ relative to the camera satisfies:

$$\frac{d\mathbf{R}}{dt} = -\mathbf{t} - (\omega \times \mathbf{R}) \tag{4}$$

Instantaneously Constant Scene Brightness

A common method used to relate the apparent motion of image points to the measured brightness $E(x, y)$ is through the constant brightness assumption. We assume that the brightness of a surface patch remains constant as the camera moves, implying that the total derivative of brightness is zero:

$$\frac{dE}{dt} = E_t + \nabla E \cdot \frac{d\mathbf{r}}{dt} = 0 \tag{5}$$

$$E_t = \frac{\partial E}{\partial t}, \qquad \nabla E = (E_x, E_y, 0)^T = \left(\frac{\partial E}{\partial x}, \frac{\partial E}{\partial y}, 0\right)^T$$

In practice, the constant brightness assumption has been shown to be valid for a large class of image sequences (Horn and Schunk, 1981).

By differentiating the perspective change equation and substituting both the rigid body and constant brightness assumptions, we find the general brightness-change constraint equation (BCCE) for both translation and rotation:

$$E_t + \frac{\mathbf{v} \cdot \omega}{f} + \frac{\mathbf{s} \cdot \mathbf{t}}{\mathbf{R} \cdot \hat{\mathbf{z}}} = 0 \tag{6}$$

The $\mathbf{s}$ and $\mathbf{v}$ vectors are strictly properties of the image brightness gradients along with the x and y position in the image:

$$\mathbf{s} = \begin{bmatrix} -fE_x \\ -fE_y \\ xE_x + yE_y \end{bmatrix}$$
$$\mathbf{v} = \begin{bmatrix} +f^2E_y + y(xE_x + yE_y) \\ -f^2E_x - x(xE_x + yE_y) \\ f(yE_x - xE_y) \end{bmatrix} \tag{7}$$

In order to investigate the case for translation only, we set the camera rotation ω to zero and define the FOE, $\mathbf{r}_0$, as the intersection of the translational velocity vector $\mathbf{t}$ with the image plane. Using the perspective projection equation, we find:

$$\mathbf{r}_0 = (x_0, y_0, f) = \frac{f\mathbf{t}}{\mathbf{t} \cdot \hat{\mathbf{z}}} \tag{8}$$

Simplifying the constraint equation in this case results in:

$$E_t + \frac{t_z}{Z}(\nabla E \cdot (\mathbf{r} - \mathbf{r}_0)) = 0 \qquad (9)$$

and rewriting this gives our final result for the BCCE under translation only:

$$\tau E_t + (x - x_0)E_x + (y - y_0)E_y = 0 \qquad (10)$$

The time-to-impact $\tau = Z/t_z$ is the ratio of the depth to the velocity parallel to the optic axis. This is a measure of the time until the plane parallel to the image plane and passing through the center of projection intersects the corresponding world point. The time-to-collision of the camera is the time-to-impact at the focus of expansion.

Examining Eq. (10), we note that the time-to-impact τ is a function of x and y, while the FOE is a global parameter. Given a time-to-impact map or assuming a special form of scene geometry such as a plane (Negahdaripour and Horn, 1986), the problem is overdetermined and we can recover the FOE using a least-squares minimization approach. However, we were interested in the more general case where the time-to-impact map and the motion are both unknown. In this situation, the problem is underdetermined and a more creative method must be found.

3. Two Algorithms Suitable for Analog VLSI

The paramount consideration that we used when examining algorithms for estimating the FOE was the feasibility of implementing them in analog VLSI. Adhering to the focal-plane processing approach in the analog domain necessitates algorithms which use low-level computations operating locally and in parallel on image brightness. Furthermore, in order to enhance the likelihood of actual implementation, any algorithm we propose must be exceedingly simple. The BCCE gives a useful low-level relationship between the location of the focus of expansion in the image plane and the observed variation of image brightness, and we would like to exploit this to estimate the FOE. Unfortunately, this relation also includes the unknown time-to-impact τ. In order to still use the BCCE without explicit knowledge of τ, two approaches were proposed. In the first approach, image points where the time variation of the brightness, E_t, is zero are identified. Ideally, the FOE would be at the intersection of the tangents to the iso-brightness contours at these so-called "stationary" points. In the second approach, the observation is made that when given an incorrect estimate of the location of the FOE, solving the BCCE for τ gives rise to depth estimates with incorrect sign. However, depth is positive and thus an estimate for the FOE can be found which minimizes the number of negative depth values.

3.1. The Stationary-Points Algorithm

Image points where $E_t = 0$ provide important constraints on the direction of translation; they are referred to as stationary points (Horn and Weldon, 1988). With $E_t = 0$, the first term of the BCCE drops out and the constraint at the stationary points becomes one of orthogonality between the measured **s** and the translation vector **t**:

$$\mathbf{s} \cdot \mathbf{t} = 0 \qquad (11)$$

Previous approaches utilized these special constraints to estimate **t** directly as opposed to finding the FOE. A least-squares minimization sum over the stationary points can be formed with an additional term utilizing a Lagrange multiplier to insure that the magnitude of **t** is normalized to unity. This normalization is necessary in order to account for the inherent scale factor ambiguity in **t**. The solution to this minimization problem is itself an eigenvector/eigenvalue problem: the estimate for **t** which minimizes the sum is the eigenvector corresponding to the smallest eigenvalue (Negahdaripour and Horn, 1987a; Horn and Weldon, 1988). Calculation of eigenvectors and eigenvalues in analog hardware is possible, but difficult (Horn, 1990).

To find a solution more amenable to implementation, we can instead perform a similar minimization now in terms of the FOE, and this leads to a simple linear problem. The constraint at each stationary point becomes a line:

$$\nabla E \cdot (\mathbf{r} - \mathbf{r}_0) = (x - x_0)E_x + (y - y_0)E_y = 0$$

Figure 3 demonstrates the simple geometry of a stationary point. For illustrative purposes, we have constructed a Mondrian image consisting of a dark circular disk on a white background. As such, the image brightness gradient ∇E points everywhere outward from the disk. A stationary point occurs on the disk when the brightness gradient is perpendicular to the vector emanating from the FOE. The focus is located at the intersection of the tangents to the brightness gradient at these points.

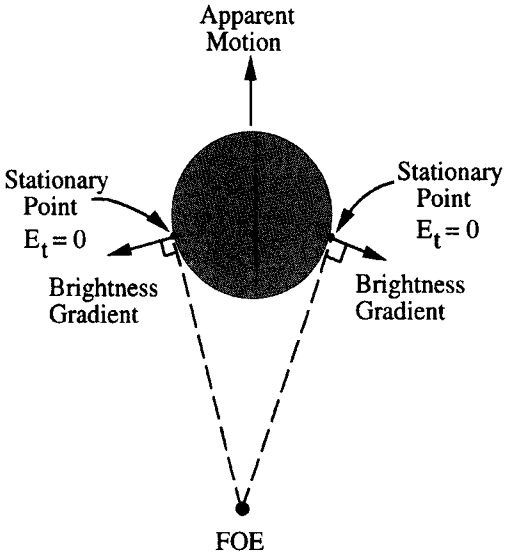

Figure 3. An illustration of the simple geometry in the image plane associated with stationary points using a Mondrian image consisting of a dark disk. The FOE is located at the intersection of the tangents at the stationary points.

Of course, in a real image these tangent lines will not precisely intersect. In such a case, we can then find a solution by minimizing the sum of the squares of the perpendicular distances from $\mathbf{r}_0$ to these constraint lines, weighted by the norm of the brightness gradient $\|\nabla E\|$:

$$\min_{\mathbf{r}_0} \sum_{\mathbf{r}\in I} W(E_t)(\nabla E \cdot (\mathbf{r} - \mathbf{r}_0))^2 \qquad (12)$$

Here the sum is over the entire image I and a weighting function $W(E_t)$ is used to allow only the contributions of those constraints that are considered to correspond to stationary points. Clearly, $W(\cdot)$ is constructed to weigh information more heavily at image points where $E_t \approx 0$. The additional weighting by the brightness gradient magnitude is useful, because both noise and distortion reduce our confidence in the direction of ∇E when $\|\nabla E\|$ is small.

The solution $\mathbf{r}_0$ to this quadratic minimization problem satisfies the matrix equation:

$$\left[\sum_{\mathbf{r}\in I} W(E_t)(\nabla E \nabla E^T)\right]\mathbf{r}_0 = \sum_{\mathbf{r}\in I} W(E_t)(\nabla E \nabla E^T)\mathbf{r} \qquad (13)$$

Note that if the direction of the brightness gradient at the stationary points is uniformly distributed, then the condition number of the matrix premultiplying $\mathbf{r}_0$ is near unity and hence the solution of the matrix equation well-behaved.

It is important to observe that the actual functional form of the weighting with E_t is not essential, as long as the weight is small for large E_t and large for small E_t. Thus, in practice we are able to use a simple function such as a cutoff on the absolute value of E_t:

$$W(E_t, \eta) = \begin{cases} 1 & \text{if } |E_t| < \eta \\ 0 & \text{otherwise} \end{cases} \qquad (14)$$

where the width of the function is set by the parameter η.

Posing the problem in terms of the FOE leads to a simple linear equation and it is this which is quite appealing for realization in analog VLSI. However, a drawback to this approach is that our estimate of the FOE relies on information garnered from the stationary points, and these points usually form a small subset of the overall image. The small number of points contributing to the solution as well as the selection of these points via the weighting function raises the question of noise immunity.

It is also important to note that the algorithm can fail if the range of τ is too large. The conflict here is in differentiating between stationary points which constrain the location of the FOE and stationary points due to distant backgrounds which do not. For example, all points along a horizon in the image have $Z \to \infty$ resulting in $E_t \approx 0$ even though they need not satisfy $\nabla E \cdot (\mathbf{r} - \mathbf{r}_0) = 0$. In such a situation, the solution will be strongly biased by these points, all of which will be selected by the weighting function to contribute to the solution. This difficulty, which is characteristic of all methods utilizing information obtained from stationary points, can certainly be alleviated using more sophisticated processing. However, this kind of higher-level understanding is outside our single-chip analog VLSI framework, and hence was eliminated from consideration.

3.2. The Depth-is-Positive Algorithm

The depth-is-positive approach was formulated in an attempt to address the problems associated with the stationary-points algorithm. This method for estimating the FOE is based on the idea that the depth calculated from the BCCE with the correct location of the FOE should be positive (Negahdaripour and Horn, 1987b). Since the BCCE only involves the ratio of depth to forward velocity, there is an overall sign

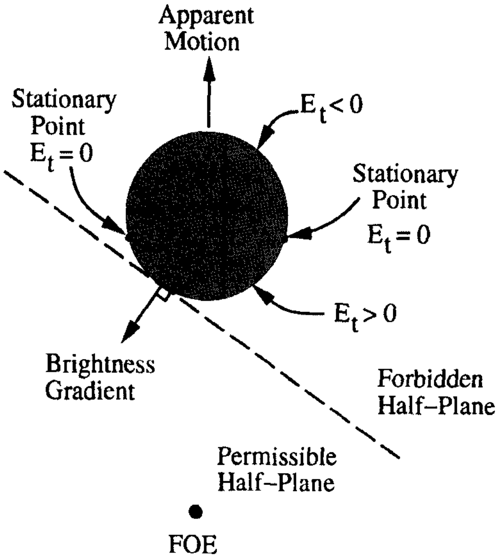

Figure 4. An illustration of the constraint provided by imposing positive depth at an image point. The tangent to the image brightness divides the image plane into permissible and forbidden half-planes.

ambiguity since this velocity can be either positive or negative and in the latter case the focus of expansion would become a focus of constriction. However, if we assume a priori that we have forward motion then we can require that the estimated τ found by solving the BCCE be positive:

$$\operatorname{sign}(\tau) = \operatorname{sign}(E_t \nabla E \cdot (\mathbf{r}_0 - \mathbf{r})) > 0$$

Returning to our simple Mondrian image, Fig. 4 illustrates the constraint line found by imposing positive depth. For each image point, the tangent to the brightness gradient can be drawn. If the FOE estimate is placed on one side of this line, τ will evaluate positive, while on the other side it will evaluate negative. Hence, each image point constrains the FOE to lie in a permissible half-plane and the true FOE must therefore lie in the region formed by the intersection of all the permissible half-planes of the points in the image. Each constraint provided by imposing positive depth is weaker than that provided by a stationary point. However, the depth-is-positive constraint applies at all image points with nonzero brightness gradient ∇E, not just a select few where $E_t = 0$, and this observation holds out the possibility that our solution can potentially rely on substantially more points and may be more robust as a consequence.

To cast the depth-is-positive constraint in terms of a minimization problem, we can formulate an error sum using only the sign of the calculated depth.

$$\min_{\mathbf{r}_0} \sum_{\mathbf{r}\in I} u(-\tau(\mathbf{r}_0)) = \min_{\mathbf{r}_0} \sum_{\mathbf{r}\in I} u(E_t \nabla E \cdot (\mathbf{r} - \mathbf{r}_0)) \qquad (15)$$

where $u(t)$ is the unit step function. The solution to this problem attempts to find the location of the FOE that minimizes the number of image points which give negative depth values. This is a difficult problem to solve since the sum is not convex. To ameliorate this difficulty, we can attempt to include some convexity in addition to the sign information in the minimization sum. To motivate the form of this sum, we can make the following observation. If we use an incorrect value for the FOE of $\mathbf{r}'_0$ and solve for the resulting τ' given by the BCCE, we find that:

$$\frac{\tau'}{\tau} = \frac{\nabla E \cdot (\mathbf{r} - \mathbf{r}'_0)}{\nabla E \cdot (\mathbf{r} - \mathbf{r}_0)}. \qquad (16)$$

Hence, not only can we get negative depth values for an incorrect FOE location, but they can also be large in magnitude. Thus it seems reasonable to augment the error sum of Eq. (15) to:

$$\min_{\mathbf{r}_0} \sum_{\mathbf{r}\in I} (E_t \tau(\mathbf{r}_0))^2 \, u\,(-\tau(\mathbf{r}_0)) = \min_{\mathbf{r}_0} \sum_{\mathbf{r}\in I} (\nabla E \cdot (\mathbf{r} - \mathbf{r}_0))^2 \, u\,(E_t \nabla E \cdot (\mathbf{r} - \mathbf{r}_0)), \qquad (17)$$

where not only do we attempt to minimize the number of negative depth values, but we also attempt to minimize their magnitudes as well. The explicit weighting with E_t in the left-hand side of Eq. (17) is intended to enhance noise robustness, since we naturally have a higher confidence in our measurements if E_t is large. However, for the case of a distant background, nonidealities such as camera noise will give small nonzero values of E_t for many pixels where in principle $E_t = 0$ and a substantial fraction of these will introduce erroneous terms into the right-hand side of Eq. (17). The conflict here is that in the low-noise case we especially value constraints from points where $E_t = 0$ as is evidenced by the stationary-points algorithm itself, but in the high-noise case small values of measured E_t may come from the distant background and give negative depth values inappropriately.

Note the partial similarity in form between the stationary points Eq. (12) and the depth-is-positive

Eq. (17). Clearly, any even power in this sum would suffice to give the desired convexity. Hence, the choice of a quadratic is rather arbitrary and in fact is motivated solely by its simplicity, a necessary feature from our implementation standpoint.

To see that the sum in Eq. (17) is convex, recall that the sum of convex functions is itself a convex function (Fleming, 1965). Each term in (17) consists of the product of a rank-1 quadratic form on the plane and a function which takes the two values zero and unity, with the transition between zero and unity occurring along the line where the quadratic form vanishes. Thus, the graph of each term vanishes on the half-plane (or everywhere for the special case $E_t = 0$) and is quadratic on the other half-plane. Each term is therefore convex (though not strictly convex) and thus the sum must be as well. Note that each term is also continuously differentiable, and thus the sum is also continuously differentiable.

Though each term is not strictly convex, the sum usually is. To see this note that the sum of a set of scalar functions with positive semidefinite Hessian matrices is a function with a Hessian matrix which is positive definite everywhere except at points where some nonzero vectors lie in the null space of the Hessian matrix of every term in the sum, generally a sparse or empty set of points.

Since the minimization sum is convex, any local minimum is a global minimum. Of course, the final estimate for the location of the FOE must be found iteratively, as there is no closed-form solution. This is not necessarily a drawback since a system implementing this minimization can utilize an iterative feedback loop to find the solution.

3.3. *Algorithm Performance*

In order to compare the performance of our two approaches, we took raw image data during a motion transient and processed it with each algorithm. Figure 5 shows a sample series of images taken from the 64 x 64 embedded imager on the FOE chip during a motion transient. A simple scene consisting of a grid of black disks on a white background was constructed, resulting in Mondrian-like images such as the ones that we have described. Camera motion was always forward towards the target with the orientation of the camera

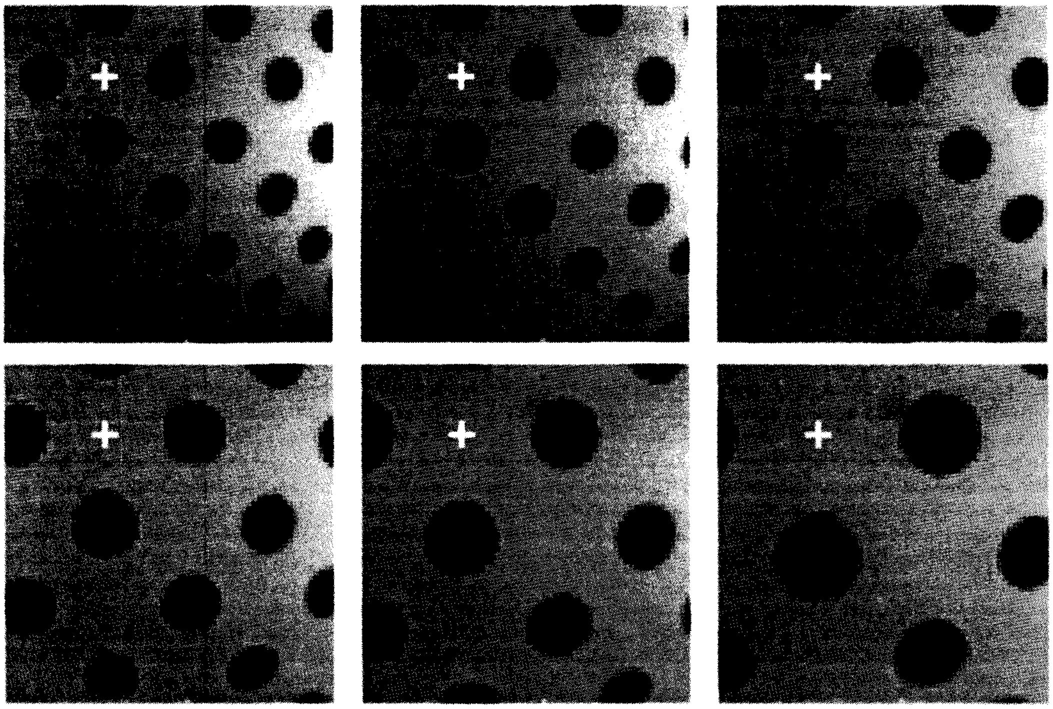

Figure 5. A sample motion sequence with the FOE placed in the upper-left hand corner as indicated by the cross. Starting at the top-left, the sequence in the figure proceeds from left to right.

viewing direction relative to the motion set precisely via rotation stages. This allowed the placement of the FOE anywhere inside the field of view.

Of course, the mapping from the 3D motion produced in the lab and the resulting location of the FOE requires explicit knowledge of the camera parameters, most notably the location of the principal point in the image plane as well as the principal distance. These were obtained using an internal camera calibration technique based on rotation (Stein, 1993). Under pure rotation, the position of a point in the image after rotation depends only on the camera parameters and the location of the point in the image before rotation. In order to estimate these parameters, a series of images for various rotations of the camera about two independent axes were taken. Feature detection to locate the centers of the disks in the images was performed and the resulting correspondences from unrotated to rotated images noted. This correspondence information was then fed to the nonlinear optimization code of (Stein, 1993) which estimates the principal distance f, the location of the optic axis (c_x, c_y), as well as the radial distortion parameters (K_1, K_2) and the axes of rotation used. Table 1 shows typical calibration parameters for the FOE chip found using this method.

Table 1. Summary of FOE chip camera calibration parameters.

Imaging parameter	Calibrated value
f	79.86 pixels
c_x	31.33 pixels
c_y	33.39 pixels
K_1	5.96E$-$5/pixel^2
K_2	1.03E$-$8/pixel^4
ϕ	0.86°

With these parameters, we can predict the location of the FOE in the image plane. A series of experiments was performed wherein the FOE was placed in a grid across the image plane and 24 frames of raw image data were acquired during each of the associated motion transients. Figure 6 shows the results of both the stationary-points algorithm and the depth-is-positive

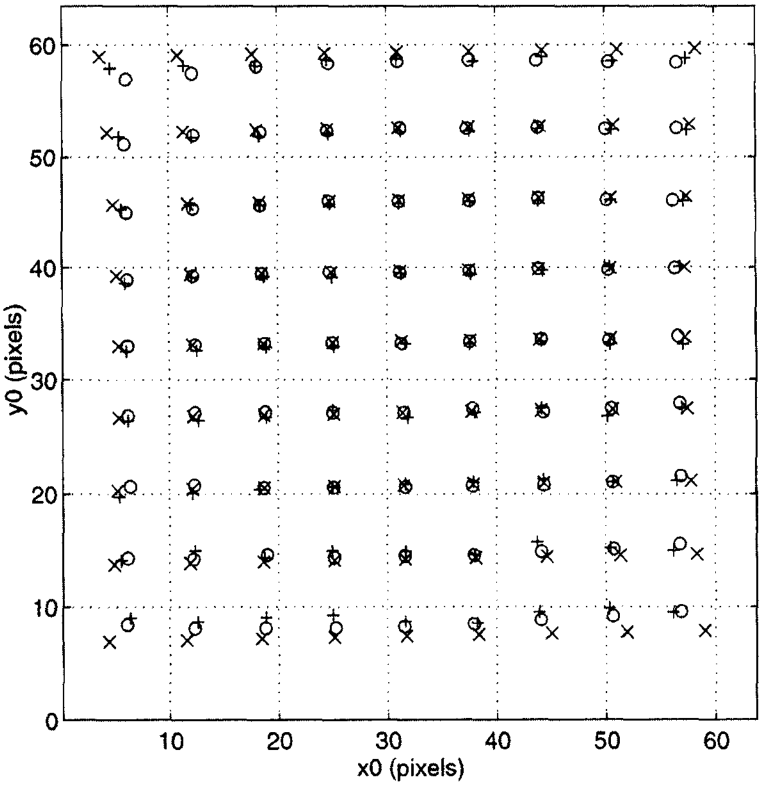

Figure 6. Algorithmic results using real image data. The actual FOE was strobed over the image plane; its position is indicated by '×'. The results of processing by the stationary-points algorithm is shown as 'o' and the depth-is-positive algorithm by '+'.

algorithm. First centered differencing was used to estimate the brightness gradients E_x, E_y, and E_t. Of course, because we have Mondrian images the regions where the brightness gradient ∇E is nonzero naturally occur only on the boundary of the disks in the image. In fact, the majority of the image has gradients near zero, and in order to prevent these from strongly biasing the solution, a threshold on the image brightness gradient $E_x^2 + E_y^2$ was used in practice to segment these out of the computation.

For estimating the location of the FOE using the stationary-points algorithm, the solution of Eq. (13) obtained by matrix inversion was utilized and for estimation using the depth-is-positive algorithm, an iterative Newton's method was employed. The locations predicted by the calibration are denoted by '×', while the mean locations of the FOE estimate over the image sequences found using the stationary-points algorithm are shown as 'o' and the depth-is-positive algorithm are shown as '+'. Both techniques produce good results near the optic axis. However, as the location of the FOE nears the image boundary, the error in the estimation increases dramatically. A variety of effects come into play when the FOE is near the image boundary, typically resulting in a deviation of the estimate towards center.

In the stationary-points algorithm, the simple constant threshold of the weighting function biases the solution to favor constraint lines due to stationary points near the FOE. The purpose of this function is to select pixels about each stationary point to contribute to the solution and hence these pixels form bands about the stationary points in the image. In general, the overall range of the time derivative E_t increases the further away from the FOE a feature is. However, E_t is still zero when $\nabla E \cdot (\mathbf{r} - \mathbf{r}_0) = 0$. Thus, E_t must go through zero more rapidly for the more distant stationary points. This effect is further exacerbated by the forshortening of image features away from the FOE. As a result, the cutoff function selects fewer points for inclusion in a band the further away from the FOE its corresponding stationary point is. Hence, bands nearby the FOE have more pixels contributing to the error sum than bands further away, and thus are effectively given more weight.

Noise will alter the directions of the measured image gradient and this angular error results in a rotation of the constraint line. For a given angular error in the gradient, the further away from the FOE the stationary point is, the larger the resulting distance between the FOE and the constraint line. Hence, the constraint information provided by distant stationary points is more noise sensitive than that given by nearby ones, and so the increased weighting of constraints from nearby stationary points is in fact a desirable effect.

The most significant contribution to the observed bias in the stationary-points algorithm is due to these bands. Each stationary point by itself should only provide a 1D constraint. However, inclusion of points in a band about a stationary point augments this constraint. Figure 7 shows the simple geometry of a band about the stationary point.

Overall, we would ideally like each band to behave as a single constraint line given by its stationary point; only distance perpendicular to this line would contribute to the error sum. However, each point in the band provides a constraint line and clearly the least-squares solution, when we neglect the contributions of the other bands in the image, falls inside the region bounded by the band itself and the constraint lines

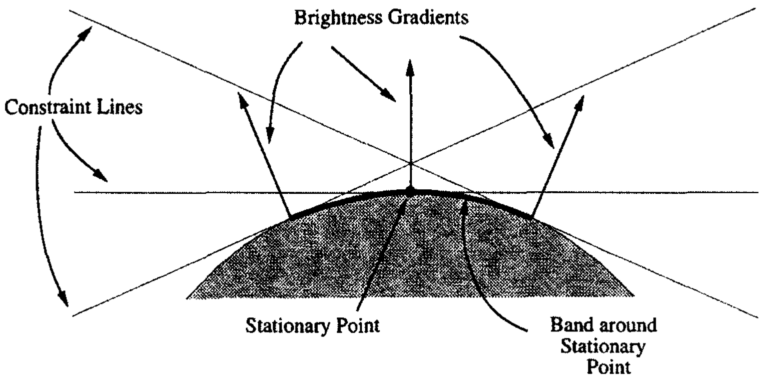

Figure 7. Simple geometry of a band about a stationary point.

provided by the points at the band edges. In effect, each band not only penalizes perpendicular distance to the effective constraint line as desired, but also the distance away from the stationary point along the constraint line. In practice, the image data had bands placed evenly about the center of the image, and so the attraction of the bands tended to cancel when the FOE was near the center. When the FOE is placed away from the center, the bands are no longer evenly placed around the FOE due to the finite extent of the image and their combined attraction draws the solution towards the center.

The depth-is-positive algorithm also displays band-like structure. The original conception of this approach was to rely on data away from the stationary points to form a solution in order to enhance robustness with respect to noise and distant backgrounds. This turns out to not be the case, as stationary points are indeed crucial to the depth-is-positive algorithm as well. If we use a test location $\mathbf{r}_0'$ for the FOE which is different than the true location and observe for what $\mathbf{r}$ negative depth values actually occur, we find that they cluster about the stationary points in bands as shown in Fig. 8. As the test location approaches that of the actual FOE, the widths of these bands shrink about the stationary points. In practice, as the noise power in the image increases, so does the effective size of the bands and hence the overall bias.

We would like to make a quantitative comparison of the performance of the two approaches in order to choose one for implementation. From a circuit design standpoint and given the architecture chosen for the FOE chip, the complexity in terms of transistor count for the two methods turned out to be roughly the same, and thus was insufficient to choose one or the other for realization in hardware. Comparing the two algorithms solely using the data of Fig. 6 can be misleading. For the stationary-points algorithm, larger widths in the weighting function lead to more data contributing to the overall solution and hence increased robustness. On the other hand, we have seen that the bands selected around the stationary points by the weighting function with larger widths lead to more bias. Thus the selection of the width of the function embodies a tradeoff between robustness and accuracy. In order to examine this tradeoff more quantitatively, synthetic images were generated which closely match the measured ones so that we could explicitly corrupt the images E with additive white Gaussian noise n to get the resulting noisy images E':

$$E' = E + n \tag{18}$$

We define the signal-to-noise ratio SNR as

$$\mathrm{SNR} = 10 \log_{10}\left(\frac{\sigma_E^2}{\sigma_n^2}\right) \tag{19}$$

where we have used the sample variance. The performance metric we constructed for purposes of comparison penalizes both bias in the solution as well as degradation due to noise. In practice, we summed the squared error between the predicted location found by the calibration technique and the mean location found by each algorithm along with the noise variance in the solution. Since the bias is spatially dependent, we averaged the results over the 81 locations in the image plane of the FOE used in our experiments, resulting in an overall metric δ intended to quantify algorithm performance:

$$\delta = \sqrt{\frac{1}{81} \sum_{\{\mathbf{r}_0\}} \left[\|\mathbf{r}_0 - \bar{\mathbf{r}}_0\|^2 + \sigma_{\mathbf{r}_0}^2 \right]} \tag{20}$$

For the stationary-points algorithm, δ is obviously a function of both the signal-to-noise ratio and the weighting function width η. However, δ shows a marked minimum with respect to η, and hence we can find the optimal width to use in practice. Figure 9 shows δ for the depth-is-positive algorithm and the optimal δ for the stationary-points algorithm, both as a function of SNR.

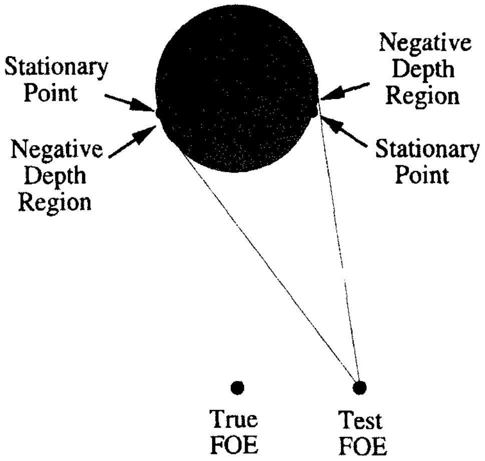

Figure 8. When a test FOE differing from the true FOE is used to calculate τ, the resulting negative depth values are clustered around the stationary points.

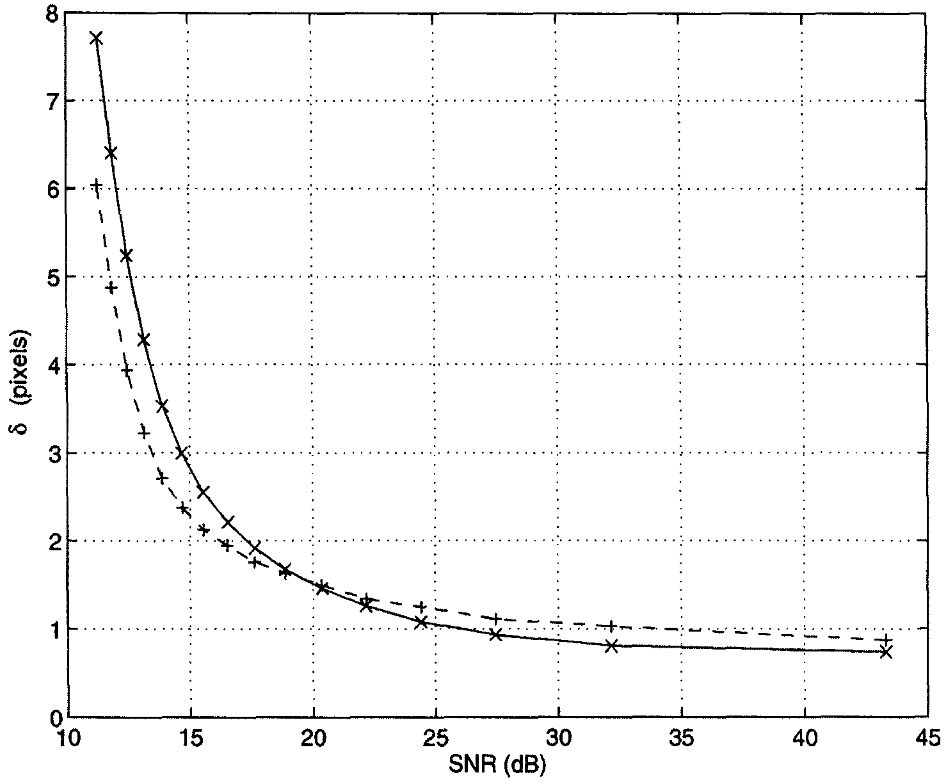

Figure 9. Algorithmic results using synthetic image data comparing the performance of the stationary-points algorithm (×) versus the depth-is-positive algorithm (+).

One important feature that one should note from these two curves is that δ does not go to zero even in the limit of noise-free image data. The reason for this is two-fold. Bias always remains, especially with the FOE near the image edge, even in the absence of noise. Furthermore, the imaging and finite-differencing process itself results in a variation in the estimated image gradient directions during a motion transient. The contribution due to this variation is always present, and can be thought of as an equivalent "noise" source.

For large signal-to-noise ratios, the optimal stationary-point behavior is slightly better than depth-is-positive, while for small signal-to-noise ratios the converse is true. Overall, the curves appear markedly similar, and do not in and of themselves provide a definitive means for choosing one algorithm over the other for implementation. However, due to the flexibility in explicitly setting the accuracy versus noise robustness tradeoff and because typical SNRs observed from the FOE chip were in the 40 dB range, the stationary-points algorithm was chosen for implementation.

4. The FOE Chip

Having chosen the stationary-points algorithm for implementation, one could design a system to calculate the individual elements of the 2 × 2 linear system of Eq. (13) and solve for the location of the FOE off-chip by matrix inversion. This would require the on-chip calculation of five complex quantities over the entire image and this makes such an approach prohibitively expensive. Instead, we can design a system to estimate the location of the FOE using a feedback technique such as gradient descent. By using this kind of approach, we can trade off the complexity of the required circuitry with the time required to perform the computation.

Given a convex error function $f(\alpha)$ of a parameter vector $\alpha = (\alpha_0, \ldots, \alpha_{N-1})^T$, we can minimize f via the set of differential equations:

$$\frac{d\alpha}{dt} = -\beta \nabla_\alpha f(\alpha), \tag{21}$$

where β is a positive definite matrix. This system of differential equations will relax to a local minimum of $f(\alpha)$ if one exists.

The stationary-points minimization sum of Eq. (12) is convex. In fact, unless all the ∇E terms with nonvanishing W are exactly parallel, then the sum is strictly convex and hence cannot have any local minimum other than the global minimum. Applying gradient descent to our particular problem results in:

$$\frac{d\mathbf{r}_0}{dt} = \beta \sum_{\mathbf{r}\in I} W(E_t, \eta)\nabla E \nabla E^T(\mathbf{r} - \mathbf{r}_0) \qquad (22)$$

To implement this system, one could use the approach of (Tanner and Mead, 1986) and design a pixel-parallel chip consisting of an $n \times n$ array of analog processors. With a photo-transistor as the imaging device, each processor would estimate the brightness gradient in time using a differentiator, and the brightness gradient in space using finite differencing with adjacent pixel processors. Based on these measured image gradients, the processor at position (x, y) in the array would calculate two currents proportional to the term inside the summation of Eq. (22). Each processor then injects these currents into global busses for the voltages x_0 and y_0, respectively, thereby accomplishing the required summation over the entire image. For a capacitor, the derivative of the voltage is proportional to the injected current, so if we terminate the busses with capacitances C_x and C_y, we naturally implement Eq. (22).

The major difficulty with this elegant solution is that of area. The output currents that the processors calculate require four multiplies in addition to the cutoff weighting function. Including all of this circuitry in addition to the photo-transistor creates a very large pixel processor size and, given the constraints of limited silicon area, the number of pixels that we would be able to put on a single chip would be quite small as a result. The actual number of pixels contributing to our computation is already small to begin with because the number of stationary points in the image is typically only a fraction of the total number of pixels in the image. A large number of pixels is clearly desirable to enhance the robustness of the computation. Additionally, a fully parallel implementation would be inefficient, again because only a small number of processors would be contributing at any one time, with the rest idle.

To increase the number of pixels and make more efficient use of area, the solution that was decided upon was to multiplex the system using a column-parallel processing scheme. Instead of computing the full frame of terms in our summation in parallel, we calculate a column of them at a time, and process the column sums sequentially. Of course, we can no longer use the simple time derivative in the right-hand side of Eq. (22). We can instead use a forward difference approximation, resulting in a simple proportional feedback system:

$$\mathbf{r}_0^{(i+1)} = \mathbf{r}_0^{(i)} + h \sum_{\mathbf{r}\in I} W(E_t, \eta)\nabla E \nabla E^T\big(\mathbf{r} - \mathbf{r}_0^{(i)}\big),$$

where h is the feedback gain. This system is now a discrete-time analog system as opposed to the continuous-time analog system discussed earlier. This implementation method will allow us to put more pixels on the chip at the expense of taking longer for the iteration to converge to the desired solution. It is interesting to note that if we had implemented the depth-is-positive algorithm instead of the stationary-points algorithm, then the expression in the sum would merely replace $W(E_t, \eta)$ with $u(E_t \nabla E^T(\mathbf{r} - \mathbf{r}_0))$.

For the stationary-points algorithm, the equation that our system should solve is the 2×2 matrix problem:

$$A\mathbf{r}_0 = b \qquad (23)$$

where we have defined

$$A = \sum_{\mathbf{r}\in I} W(E_t, \eta)\nabla E \nabla E^T, \qquad (24)$$

$$b = \sum_{\mathbf{r}\in I} W(E_t, \eta)\nabla E \nabla E^T \mathbf{r}. \qquad (25)$$

We can rewrite our solution method into the following form:

$$\mathbf{r}_0^{(i+1)} = \mathbf{r}_0^{(i)} + h\big(b - A\mathbf{r}_0^{(i)}\big). \qquad (26)$$

This is the Richardson method, the simplest iterative technique for solving a matrix equation (Varga, 1962). The transient solution to this equation is:

$$\mathbf{r}_0^{(i)} = A^{-1}b + (I - hA)^i\,\mathbf{e}_0 \qquad (27)$$

where $A^{-1}b$ is the desired solution and $\mathbf{e}_0$ is the initial error. Clearly, in order for this system to be stable, we require that the error iterates go to zero. We must therefore guarantee that the spectral radius of the iteration matrix is less than unity. Examining the eigenvalues λ' of the iteration matrix we find that they are related to the eigenvalues λ of the matrix A by

$$\lambda' = 1 - h\lambda. \qquad (28)$$

Since A is symmetric and positive semidefinite (typically definite in practice), we know that the eigenvalues of A are real and positive. Requiring the spectral radius of the iteration matrix to be less than unity results in the following requirement on h for stability

$$0 \leq h \leq \frac{2}{\lambda_{\max}} \tag{29}$$

Additionally, we can choose the optimal h to minimize the convergence time of the iteration. This h_{opt} solves:

$$\begin{aligned} h_{\text{opt}} &= \arg\min_h(\max(|1 - h\lambda_{\min}|, |1 - h\lambda_{\max}|)) \\ &= \frac{2}{\lambda_{\min} + \lambda_{\max}} = \frac{2}{\sum_{\mathbf{r}\in I} W(E_t, \eta)\|\nabla E\|^2} \end{aligned}$$

At a minimum, we therefore require our system to calculate three quantities: the matrix residual $b - A\mathbf{r}_0^{(i)}$, the weighted squared image gradient $\sum_{\mathbf{r}\in I} W(E_t, \eta)$ $\|\nabla E\|^2$, and a fourth quantity, $\sum_{\mathbf{r}\in I} |E_t|$, useful in practice for setting the width η of the weighting function (McQuirk, 1991).

The approach that was decided upon to implement the discrete time system we have described uses charge-coupled devices (CCDs) as image sensors. If we expose a CCD to light over a short period of time, it stores up a charge packet which is linearly proportional to the incident light during this integration time. Arrays of CCDs can be manipulated as analog shift registers as well as imaging devices. This allows us to easily multiplex a system which uses CCDs. Since we intend to process image data in the voltage/current domain, we must convert the image charge to voltage and this can be done nondestructively through a floating gate amplifier. Thus, we can shift our image data out of a CCD array column-serial and perform our calculations one column at a time. Instead of n^2 computational elements corresponding to the parallelism of a continuous-time system, we now only have n. Clearly, we can increase our pixel resolution significantly and design more robust circuitry to perform the computations as a result.

The system architecture used in the FOE chip is shown in Fig. 10. (More explicit circuit data can be found in (McQuirk, 1997), and a complete discussion of FOE circuit and system performance can be found in

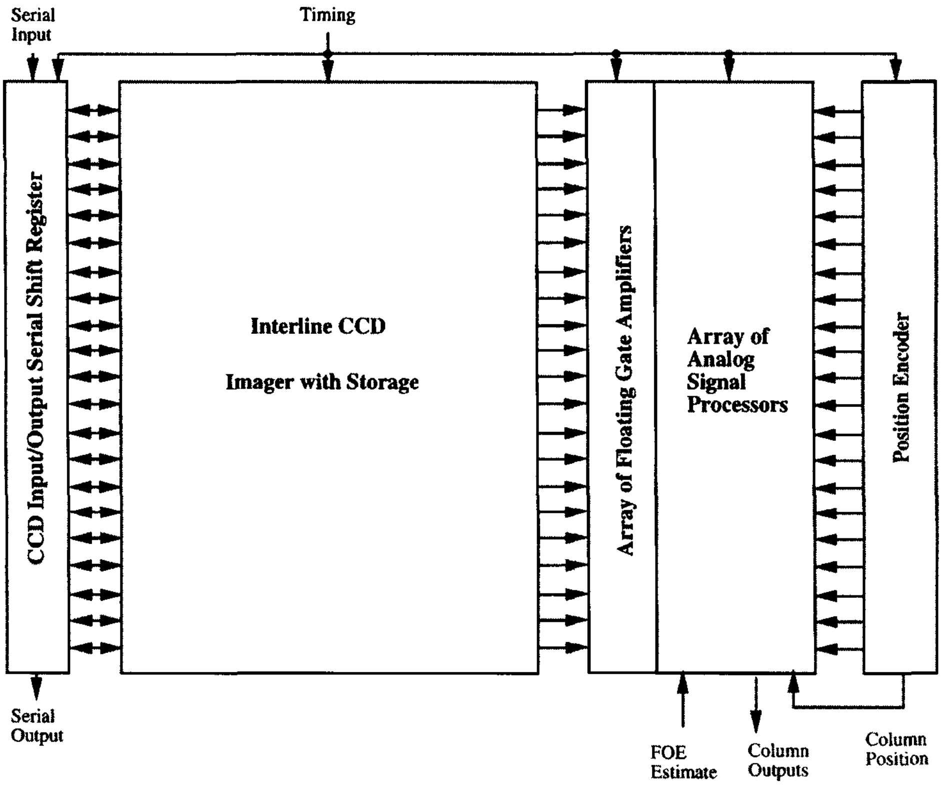

Figure 10. Block diagram representation of the system architecture of the actual FOE chip.

(McQuirk, 1996b) which is also available as (McQuirk, 1996a).) It is composed of four main sections: the CCD imager with storage and an input/output serial shift register, the array of floating gate amplifiers for transducing image charge to voltage, the CMOS array of analog signal processors for computing the required column sums, and the position encoder providing (x, y) encoding in voltage to the CMOS array as data is processed.

The input/output CCD shift register at the left side of the block diagram allows us to disable the imager, and insert off-chip data into the computation. This shift register can also clock data out of the CCD imager, letting us see the images that the system is computing with. Thus we have four basic testing modes: (i) computer simulated algorithm on synthetic data, (ii) computer simulated algorithm on raw image data taken from the imager, (iii) chip processing of synthetic data input from off-chip, and (iv) chip processing of raw image data acquired in the on-chip imager. With these four testing modes, we can separately evaluate algorithm performance and system performance.

The function of the interline CCD imager with storage is to acquire the two images in time necessary to estimate the brightness gradients. The imager is an interline structure which uses exposed photogates to integrate image charge and covered shift registers to both store the resulting images and shift them out. Once two images have been acquired successively in time, we shift them to the right one column at a time to an array of floating gate amplifiers. The floating gate amplifiers transduce the image charges into voltages which are then applied to the analog signal processors. As input, these processors also require the current estimate in voltage of the location of the FOE driven in from off-chip, $\mathbf{r}_0^{(i)} = (x_0^{(i)}, y_0^{(i)})$, and the present $\mathbf{r} = (x, y)$ position of the data, provided in voltages by the position encoder at the far right of the diagram.

The encoder uses the voltage on a resistive chain to encode the y position up the array. A CMOS digital shift register is utilized to select the appropriate x value over time as columns are processed. A logic 1 is successively shifted up the shift register as each column is processed, enabling a pass transistor which sets x to the value of voltage on the resistor chain at that stage. In this manner, x increases in the stair-step fashion necessary as the columns of data are shifted through the system.

From the image data, the pixel position, and the FOE estimate, the processors in the array compute the four desired output currents which are summed up the column in current and sent off-chip. The block diagram for the analog processors is shown in Fig. 11.

To estimate the three brightness gradients, eight input voltages representing the $2 \times 2 \times 2$ cube of pixels needed for the centered differencing are input to the processor. Four MOS source-coupled pairs are used to transduce these voltages into differential currents which are then added and subtracted using current mirroring to form the brightness gradients. An absolute

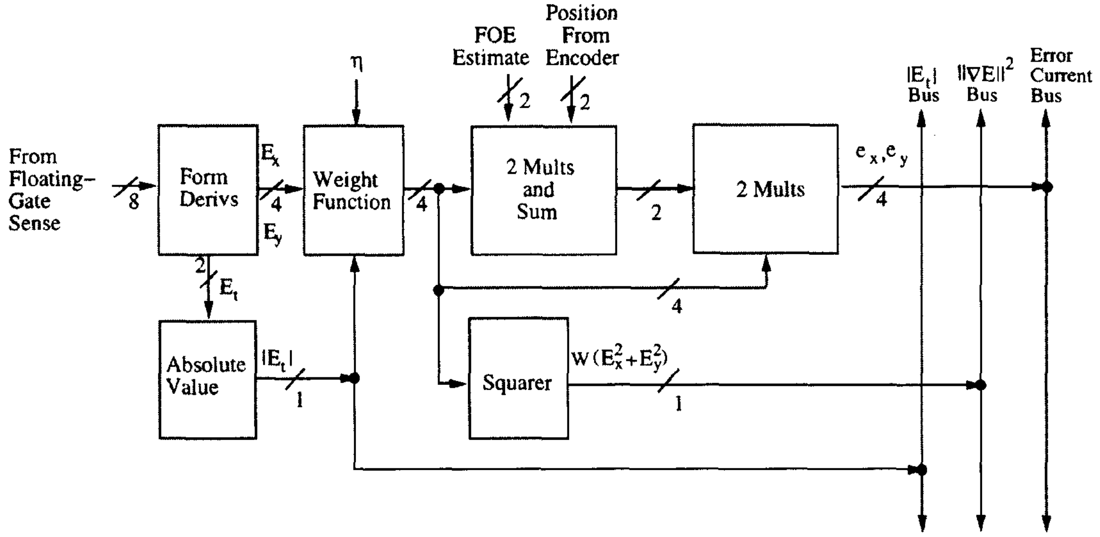

Figure 11. Block diagram indicating the structure of analog row processor in the CMOS processing array.

value circuit computes $|E_t|$ and a copy of this signal is then summed up in current along with all the contributions of the other processors in the array. The resulting overall current forms the first main output of the chip, $\sum_y |E_t|$.

Another copy of $|E_t|$ is subtracted from a reference current I_η and injected into a single-ended latch. The result of this latch is the weighting function decision. In-line with the brightness gradient currents E_x, E_y are pass-gate switches whose state is controlled by the weighting decision. If the processor is not to contribute to the error sum because $|E_t| > \eta$, these gates are turned off by the output of the latch preventing signal flow to the rest of the processor and thus enforcing the weighting decision. The weighted brightness gradients are copied using current mirrors three times—the first is used for the pair of current-mode squarers needed to compute the squared gradient magnitude. This signal is summed up in current along with all the contributions of the other processors in the array and the resulting overall current forms the second main output of the chip, $\sum_y W(E_t, \eta)\|\nabla E\|^2$.

The second gradient copy is used by the first layer of multipliers to compute $W(E_t, \eta)\nabla E \cdot (\mathbf{r} - \mathbf{r}_0)$. The multipliers used on the FOE chip are all simple MOS versions of the standard four-quadrant Gilbert bipolar multipliers (Gilbert, 1968). These multipliers have as input both a differential voltage and a differential current. The output is a differential current which approximates a multiplication of the two inputs. The dot product resulting from the first layer of multipliers is transduced from differential current to differential voltage using an MOS circuit with devices operating in the triode region. This signal is then used as the voltage input to a second layer of multipliers whose other input is the third gradient copy. The resulting two final differential currents are the contribution to the matrix equation residual for that processor. They are summed in current along with the contributions from the other processors in the array, and form the final two outputs from the chip.

To complete the iterative feedback loop, we sum the output currents from the column of analog processors as the image data is shifted out a column at a time from the imager. Once a whole frame of data has been accumulated, we use the residual to update the FOE estimate using the proportional feedback loop. While this certainly could be done on the chip, this was done off-chip in DSP for testing flexibility. Due to the difficulty in re-circulating image data on-chip, we further acquire new image pairs for each successive iteration of the feedback loop. Since the locations of the stationary points in the image move as the camera translates, using new image pairs at each iteration can be viewed as producing additional noise for the algorithm to handle. We could alleviate this problem by moving the imager off-chip and adding a frame buffer, but the original architectural goal was a single-chip system.

A series of experiments were performed in which the FOE was placed in a grid across the image plane. For testing purposes, image data corresponding to real motion was generated using a flexible fiber-optic image guide. One end of the image guide was held fixed over the chip, while the other end was attached to a carrier whose position and velocity on a linear track was controlled through a DC motor system. The orientation of the viewing direction of the image guide relative to the motion direction of the carrier was precisely set using two rotation stages, allowing for the positioning of the FOE anywhere in the image plane. Figure 12 shows the final results from the FOE chip comparing the mean output of the proportional feedback loop enclosing the chip (shown as '$*$') with the results of the algorithm on the raw image data (shown as '$\circ$').

5. Summary

This paper discussed the application of integrated analog focal plane processing to realize a real-time system for estimating the direction of camera motion. The focus of expansion is the intersection of the camera translation vector with the image plane and captures this motion information. Knowing the direction of camera translation clearly has obvious import for the control of autonomous vehicles, or in any situation where the relative motion is unknown. The mathematical framework for our approach, resulting in a simplified brightness change constraint equation, was developed. Several promising algorithms for estimating the FOE based on this constraint and suitable for analog VLSI were discussed, including the one chosen for final implementation. A special-purpose VLSI chip with an embedded CCD imager and column-parallel analog signal processing was constructed to realize the desired algorithm. The difference between the output of the FOE chip enclosed in a simple proportional feedback loop and the location predicted by the stationary-points algorithm operating on raw image data was less than 3% full scale. Table 2 summarizes the performance

Table 2. Summary of FOE chip and system performance parameters.

Process	Orbit n-well 2 μm CCD/BiCMOS
Chip dimensions	9200 μm × 7900 μm
Imager topology	64 × 64 interline CCD
Technology	Double-poly buried channel CCD
Illumination	Front-face
Image sensor	CCD gate
Charge packet size	600,000 e^-
Acquisition time	Tested to 1 ms
Quantum efficiency	30% @ 637 nm
Dark current	$\leq$10 nA/cm^2 @ 30°C
Transfer inefficiency	$\leq 7 \times 10^{-5}$ @ 500 kHz
FGA output sensitivity	2 μV/e^-
System frame rate	30 frames/s
Peak on-chip power dissipation	170 mW
System settling time	20–30 iterations typical
FOE location inaccuracy	$\leq$3% full scale over 80% of the field of view

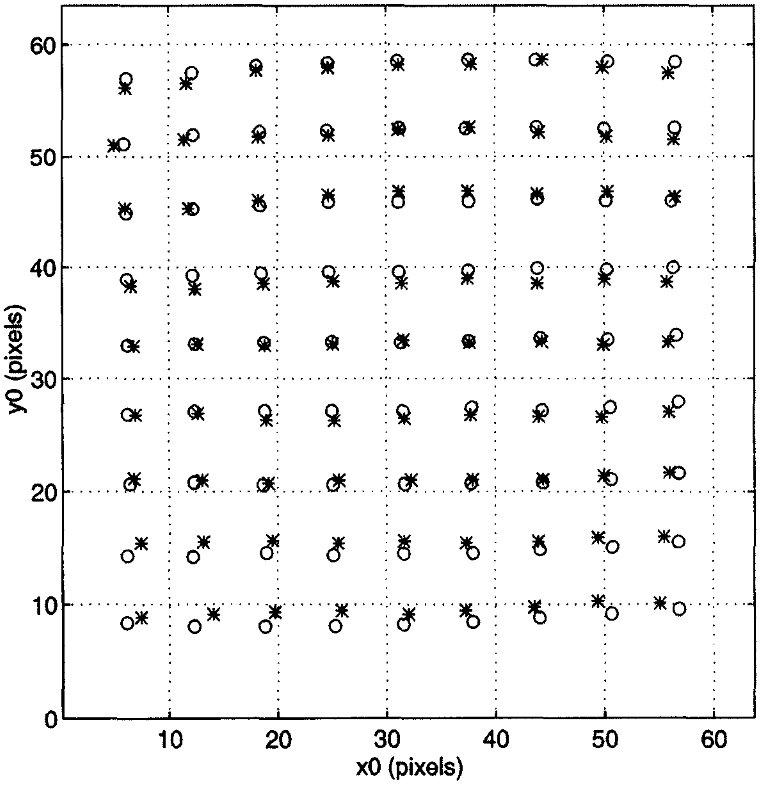

Figure 12. Comparison of the results from the stationary-points algorithm using raw image data (shown as '○') and the output from the FOE chip (shown as '∗').

achieved by the FOE chip in conjunction with the test system. A detailed discussion of FOE circuit and system performance can be found in (McQuirk, 1996b) which is also available as (McQuirk, 1996a).

Acknowledgments

The authors would like to thank Chris Umminger and Steve Decker for their many helpful comments and criticisms.

References

Bruss, A. and Horn, B. 1983. Passive navigation. *Computer Vision, Graphics, and Image Processing*, 21(1):3–20.

Dron, L. 1993. The multi-scale veto model: A two-stage analog network for edge detection and image reconstruction. *International Journal of Computer Vision*, 11(1):45–61.

Fleming, W. 1965. *Functions of Several Variables*. Addison Wesley: Reading, MA.

Gilbert, B. 1968. A precise four-quadrant multiplier with subnanosecond response. *IEEE Journal of Solid-State Circuits*, SC–3(4):365–373.

Hakkarainen, J. and Lee, H. 1993. A 40 × 40 CCD/CMOS absolute-value-of-difference processor for use in a stereo vision system. *IEEE Journal of Solid-State Circuits*, 28(7):799–807.

Horn, B. 1986. *Robot Vision*. MIT Press: Cambridge, MA.

Horn, B. 1990. Parallel networks for machine vision. In *Artificial Intelligence at MIT: Expanding Frontiers*, P. Winston and S.A. Shellard (Eds.), MIT Press: Cambridge, MA, vol. 2, chap. 43, pp. 530–573.

Horn, B. and Schunk, B. 1981. Determining optical flow. *Artificial Intelligence*, 16(1–3):185–203.

Horn, B. and Weldon, E. 1988. Direct methods for recovering motion. *International Journal of Computer Vision*, 2(1):51–76.

Jain, R. 1983. Direct computation of the focus of expansion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(1).

Keast, C. and Sodini, C. 1993. A CCD/CMOS-based imager with integrated focal plane signal processing. *IEEE Journal of Solid-State Circuits*, 28(4):431–437.

McQuirk, I. 1991. Direct methods for estimating the focus of expansion. Master's Thesis, Massachusetts Institute of Technology, Cambridge, MA.

McQuirk, I. 1996a. An analog VLSI chip for estimating the focus of expansion. AI Technical Report 1577, The MIT Artificial Intelligence Laboratory, Cambridge, MA.

McQuirk, I. 1996b. An analog VLSI chip for estimating the focus of expansion. Ph.D Thesis, Massachusetts Institute of Technology, Cambridge, MA.

McQuirk, I. 1997. An Analog VLSI Chip for Estimating the Focus of Expansion. In *1997 ISSCC Digest of Technical Papers*, pp. 24–25.

Negahdaripour, S. and Horn, B. 1986. Direct passive navigation: Analytical solution for planes. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Washington, DC, IEEE Computer Society Press.

Negahdaripour, S. and Horn, B. 1987a. Direct passive navigation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(1):168–176.

Negahdaripour, S. and Horn, B. 1987b. Using depth-is-positive constraint to recover translational motion. In *Proceedings of the Workshop on Computer Vision*, Miami Beach, FL.

Standley, D. 1991. An object position and orientation IC with embedded imager. *IEEE Journal of Solid-State Circuits*, 26(12):1853–1859.

Stein, G.P. 1993. Internal camera calibration using rotation and geometric shapes. Master's Thesis, Massachusetts Institute of Technology, Cambridge, MA.

Tanner, J. and Mead, C. 1986. An integrated optical motion sensor. In *VLSI Signal Processing II, Proceedings of the ASSP Conference on VLSI Signal Processing*, University of California, Los Angeles, pp. 59–76.

Umminger, C. and Sodini, C. 1995. An integrated analog sensor for automatic alignment. *IEEE Journal of Solid-State Circuits*, 30(12):1382–1390.

Varga, R. 1962. *Matrix Iterative Analysis*. Prentice-Hall: Englewood Cliffs, NJ.

Yang, W. and Chiang, A. 1990. A full fill-factor CCD imager with integrated signal processors. In *1990 ISSCC Digest of Technical Papers*, pp. 218–219.