MASSACHUSETTS INSTITUTE OF TECHNOLOGY
ARTIFICIAL INTELLIGENCE LABORATORY

A.I. Memo No. 1577 June, 1996

An Analog VLSI Chip for Estimating the Focus of Expansion

Ignacio Sean McQuirk

This publication can be retrieved by anonymous ftp to publications.ai.mit.edu.

Abstract

For applications involving the control of moving vehicles, the recovery of relative motion between a camera and its environment is of high utility. This thesis describes the design and testing of a real-time analog VLSI chip which estimates the focus of expansion (FOE) from measured time-varying images. Our approach assumes a camera moving through a fixed world with translational velocity; the FOE is the projection of the translation vector onto the image plane. This location is the point towards which the camera is moving, and other points appear to be expanding outward from. By way of the camera imaging parameters, the location of the FOE gives the direction of 3-D translation.

The algorithm we use for estimating the FOE minimizes the sum of squares of the differences at every pixel between the observed time variation of brightness and the predicted variation given the assumed position of the FOE. This minimization is not straightforward, because the relationship between the brightness derivatives depends on the unknown distance to the surface being imaged. However, image points where brightness is instantaneously constant play a critical role. Ideally, the FOE would be at the intersection of the tangents to the iso-brightness contours at these "stationary" points. In practice, brightness derivatives are hard to estimate accurately given that the image is quite noisy. Reliable results can nevertheless be obtained if the image contains many stationary points and the point is found that minimizes the sum of squares of the perpendicular distances from the tangents at the stationary points.

The FOE chip calculates the gradient of this least-squares minimization sum, and the estimation is performed by closing a feedback loop around it. The chip has been implemented using an embedded CCD imager for image acquisition and a row-parallel processing scheme. A 64×64 version was fabricated in a 2μm CCD/BiCMOS process through MOSIS with a design goal of 200 mW of on-chip power, a top frame rate of 1000 frames/second, and a basic accuracy of 5%. A complete experimental system which estimates the FOE in real time using real motion and image scenes is demonstrated.

This report describes research done at the Artificial Intelligence Laboratory, the Microsystems Technology Laboratory, and the Research Laboratory of Electronics at the Massachusetts Institute of Technology. Support for this research was provided in part by the National Science Foundation and the Advanced Research Projects Agency under contracts MIP-8814612 and MIP-9117724. The author was also supported by an NSF Graduate Fellowship and an AT&T Cooperative Research Fellowship.

Contents

List of Figures

List of Tables

1 Introduction

Some attention has recently been given to the potential use of custom analog VLSI chips for early vision processing problems such as optical flow [1, 2], smoothing and segmentation [3, 4, 5, 6, 7, 8] orientation [9, 10], depth from stereo [11, 12], edge detection [13, 14] and alignment [15, 16]. 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 well suited to implementation in analog VLSI, often resulting in compact, high speed, and low power solutions. These chips exploit the inherent parallelism of the computation along with the close coupling of the processing circuitry with the image sensor. This thesis details the application of this approach of focal plane processing to the early vision task of passive navigation.

An important goal of motion vision is to estimate the 3-D motion of a camera in an environment based on the resulting time-varying image sequence. Traditionally, there have been two basic approaches to this problem: *feature based methods* and *motion field based methods.* 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 [17, 13]. In motion field based methods, the apparent velocity of points in the image, the *optical flow* [18], is used to first approximate the projection of the three dimensional motion vectors into the image plane, the *motion field.* The optical flow is then used to estimate the camera motion and scene depth [19]. Both the correspondence problem and optical flow calculation have proven to be difficult in terms of reliability and implementation. We therefore look to methods which directly utilize image brightness information to recover motion [20, 21, 22, 23]. These methods lead to computations in the image domain rather than symbolic or sequential computations.

The introduction of the *focus of expansion* (FOE) for the case of *pure translation* simplifies the general motion problem significantly. 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 in Figure 1-1. If the camera velocity has a positive component along the optic axis, features will appear to be moving away from the FOE and expanding, with those close to the FOE moving slowly and those further away moving more rapidly. Through knowledge of the camera parameters, the FOE gives the direction of camera translation. Once the FOE has been determined, 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 estimate 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.

This thesis investigates the design and implementation of a chip in analog VLSI for estimation of the focus of expansion produced by camera motion. The goal of the project is to demonstrate the chip functioning in a complete motion vision system operating in real time using real motion with real image scenes. Consequently, the thesis is divided into three basic parts: a review of the algorithm chosen for implementation along with its theoretical basis, a description of the FOE chip architecture and its circuit design, and the test system and experimental setup devised to accomplish our goal.

The thesis begins in Chapter 2 by describing the algorithmic foundations for the FOE chip. The brightness change constraint equation, which forms the basis for our differential approach to the motion vision problem, is derived, allowing the formulation of a variety of low-level approaches suitable for implementation. The conclusions provided by extensive simulation of these algorithms along with a knowledge of what is feasible in analog VLSI allows us to pick one of these algorithms for implementation. Chapter 3 provides a detailed description of the structure and operation of the final architecture settled upon along with its associated circuit components. Chapter 4 describes the test system and experimental setup designed to support the chip, as well as the requirements resulting from the necessity of using real data from real motion in real time. Chapter 5 presents the test results of the FOE chip, from the circuit level to the system level to the algorithmic level, confirming its operation and characterizing its performance. Finally, the work is summarized and concluding comments made in Chapter 6.

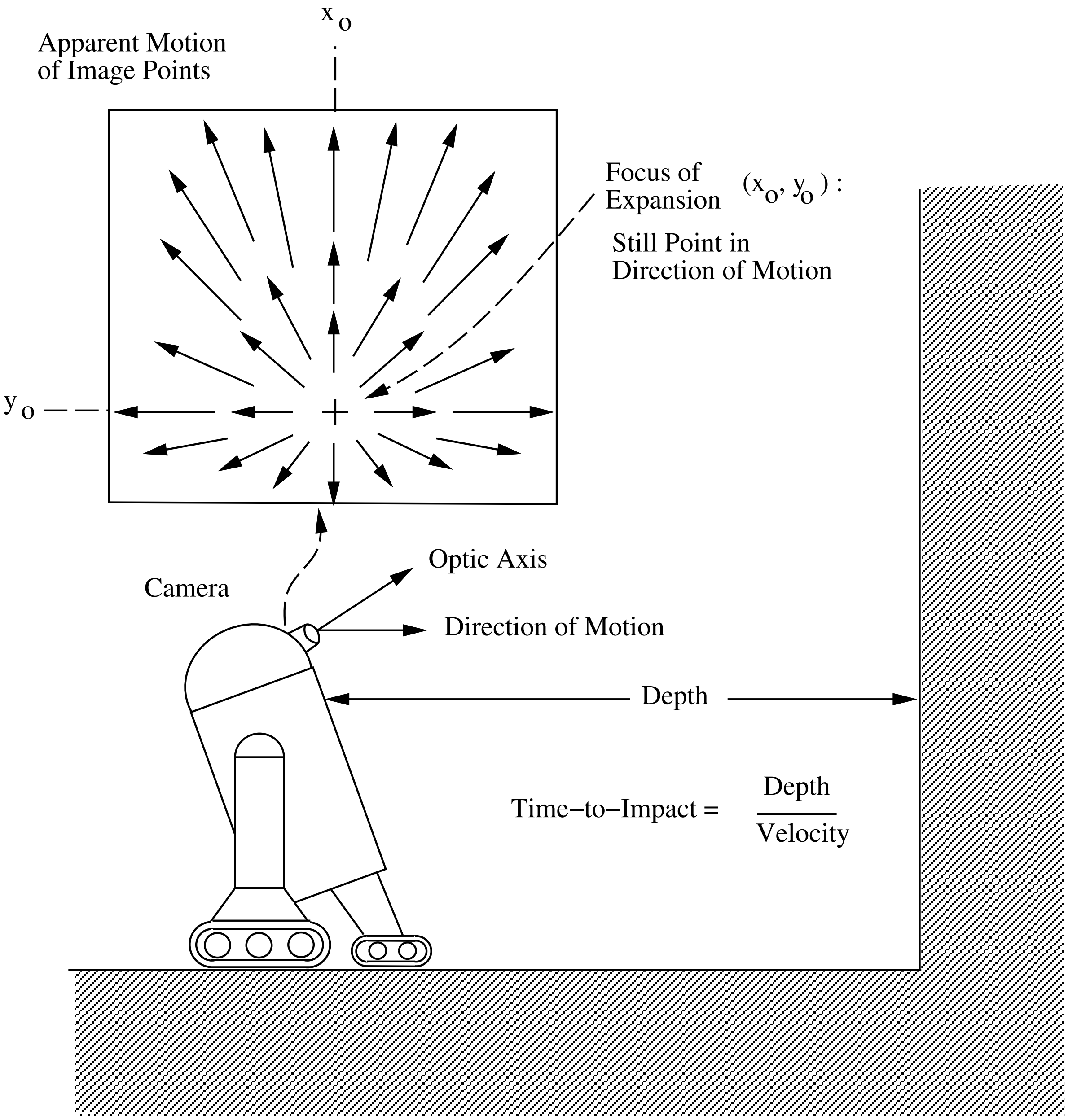


FIGURE 1-1: Illustration of the passive navigation scenario, showing the definition of the focus of expansion as the intersection of the camera velocity vector with the image plane.

2

Direct Methods for Estimating the Focus of Expansion

2.1 The Brightness Change Constraint Equation

The *brightness change constraint equation* forms the foundation of the various direct algorithms for rigid body motion vision [23, 21, 24] that we have explored for potential implementation in analog VLSI. It is derived from the following three basic assumptions:

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

Following [21, 24, 19], we use a viewer based coordinate system with a pin-hole model of image formation as depicted in Figure 2-1. A world point

$$\mathbf{R} \equiv (X, Y, Z)^T \tag{2.1}$$

is mapped to an image point

$$\mathbf{r} \equiv (x, y, f)^T \tag{2.2}$$

using a ray passing through the center of projection, which is placed at the origin of the coordinate system. The image plane $Z = f$, where f is termed the *principal distance*, is positioned in front of the center of projection. 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 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 related by projection:

$$\mathbf{r} = \frac{f\,\mathbf{R}}{\mathbf{R} \cdot \hat{\mathbf{z}}} \tag{2.3}$$

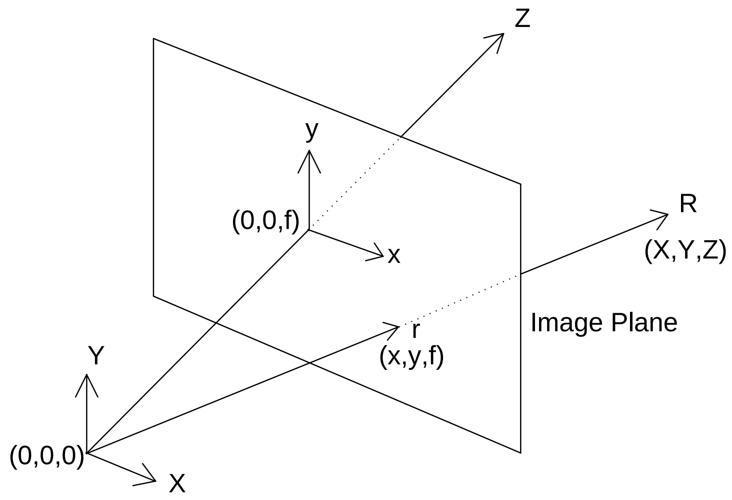


FIGURE 2-1: Viewer-centered coordinate system and perspective projection.

which is the *perspective projection equation* [25]. To find the motion of image points given the motion of world points, we differentiate Equation 2.3 using the chain rule:

$$\mathbf{r}_t = \frac{f}{(\mathbf{R}\cdot\hat{\mathbf{z}})^2}\Big((\mathbf{R}\cdot\hat{\mathbf{z}})\mathbf{R}_t - (\mathbf{R}_t\cdot\hat{\mathbf{z}})\mathbf{R}\Big) \tag{2.4}$$

where we have defined

$$\mathbf{R}_t \equiv \frac{d\mathbf{R}}{dt} = \left(\frac{dX}{dt}, \frac{dY}{dt}, \frac{dZ}{dt}\right)^T \tag{2.5}$$

$$\mathbf{r}_t \equiv \frac{d\mathbf{r}}{dt} = \left(\frac{dx}{dt}, \frac{dy}{dt}, 0\right)^T \tag{2.6}$$

We can rewrite Equation 2.4 as

$$\mathbf{r}_t = \frac{f}{(\mathbf{R}\cdot\hat{\mathbf{z}})^2}\Big(\hat{\mathbf{z}}\times(\mathbf{R}_t\times\mathbf{R})\Big) = \frac{\hat{\mathbf{z}}\times(\mathbf{R}_t\times\mathbf{r})}{\mathbf{R}\cdot\hat{\mathbf{z}}} \tag{2.7}$$

Next, we introduce the constraint implied by having the camera move relative to a fixed environment with constant translational velocity $\mathbf{t} = (t_x, t_y, t_z)^T$ and constant rotational velocity $\boldsymbol{\omega} = (\omega_x, \omega_y, \omega_z)^T$. The resulting motion of a world point $\mathbf{R}$ relative to the camera satisfies:

$$\mathbf{R}_t = -\mathbf{t} - (\boldsymbol{\omega}\times\mathbf{R}) = -\mathbf{t} - (\boldsymbol{\omega}\times\mathbf{r})\left(\frac{\mathbf{R}\cdot\hat{\mathbf{z}}}{f}\right) \tag{2.8}$$

Clearly, this constraint is valid only for *instantaneous* motion. Combining this with Equation 2.7, we arrive at a parameterization of the motion field $\mathbf{r}_t$:

$$\mathbf{r}_t = -\hat{\mathbf{z}}\times\left(\mathbf{r}\times\left(\frac{\mathbf{r}\times\boldsymbol{\omega}}{f} - \frac{\mathbf{t}}{\mathbf{R}\cdot\hat{\mathbf{z}}}\right)\right) \tag{2.9}$$

A common method to relate the motion field $\mathbf{r}_t$ to the measured image brightness $E(x,y,t)$ is to do so through the *constant brightness assumption* [23]. We assume that the brightness of a surface patch remains instantaneously constant as the camera moves. This implies:

$$\frac{dE}{dt} = E_t + \mathbf{E_r}\cdot\mathbf{r}_t = 0 \tag{2.10}$$

$$E_t \equiv \frac{\partial E}{\partial t} \ , \ \mathbf{E_r} = (E_x, E_y, 0)^T = \left(\frac{\partial E}{\partial x}, \frac{\partial E}{\partial y}, 0\right)^T \tag{2.11}$$

In practice, the constant brightness assumption is valid for a large class of image sequences [18]. Combining Equation 2.9 and Equation 2.10, we have:

$$E_t - \mathbf{E_r} \cdot \left(\hat{\mathbf{z}} \times \left(\mathbf{r} \times \left(\frac{\mathbf{r} \times \boldsymbol{\omega}}{f} - \frac{\mathbf{t}}{\mathbf{R} \cdot \hat{\mathbf{z}}}\right)\right)\right) = 0 \tag{2.12}$$

To simplify this expression, we can define the new image quantities $\mathbf{s}$ and $\mathbf{v}$ as follows:

$$\mathbf{s} \equiv (\mathbf{E_r} \times \hat{\mathbf{z}}) \times \mathbf{r}, \quad \mathbf{v} \equiv \mathbf{r} \times \mathbf{s} \tag{2.13}$$

Through the use of some vector identities, we find:

$$\mathbf{E_r} \cdot \Big(\hat{\mathbf{z}} \times (\mathbf{r} \times \mathbf{t})\Big) = \Big((\mathbf{E_r} \times \hat{\mathbf{z}}) \times \mathbf{r}\Big) \cdot \mathbf{t} \tag{2.14}$$

$$\mathbf{E_r} \cdot \Big(\hat{\mathbf{z}} \times \Big(\mathbf{r} \times (\mathbf{r} \times \boldsymbol{\omega})\Big)\Big) = -\mathbf{v} \cdot \boldsymbol{\omega} \tag{2.15}$$

Combining these results with Equation 2.12, we have:

$$E_t + \frac{\mathbf{v} \cdot \boldsymbol{\omega}}{f} + \frac{\mathbf{s} \cdot \mathbf{t}}{\mathbf{R} \cdot \hat{\mathbf{z}}} = 0 \tag{2.16}$$

This is known as the *brightness change constraint equation.*

Now, in order to investigate the case of *pure translation*, we set $\boldsymbol{\omega} = \mathbf{0}$:

$$E_t + \frac{\mathbf{s} \cdot \mathbf{t}}{\mathbf{R} \cdot \hat{\mathbf{z}}} = 0 \tag{2.17}$$

This equation remains the same if we scale both the depth Z and translational velocity vector $\mathbf{t}$ by the same scale factor. This ambiguity is referred to as the *scale-factor ambiguity.* Defining the FOE:

$$\mathbf{r}_0 = (x_0, y_0, f) \tag{2.18}$$

as the intersection of the translational velocity vector $\mathbf{t}$ with the image plane as in Figure 2-2, the following relation results:

$$\mathbf{r}_0 = \frac{f\,\mathbf{t}}{\mathbf{t} \cdot \hat{\mathbf{z}}} \tag{2.19}$$

Again using vector identities, we can note that

$$\begin{aligned} \mathbf{s} \cdot \mathbf{r}_0 &= \Big((\mathbf{E_r} \times \hat{\mathbf{z}}) \times \mathbf{r}\Big) \cdot \mathbf{r}_0 \\ &= (\mathbf{E_r} \cdot \mathbf{r})(\hat{\mathbf{z}} \cdot \mathbf{r}_0) - (\hat{\mathbf{z}} \cdot \mathbf{r})(\mathbf{E_r} \cdot \mathbf{r}_0) \\ &= f\,\mathbf{E_r} \cdot (\mathbf{r} - \mathbf{r}_0) \end{aligned} \tag{2.20}$$

Combining Equations 2.17, 2.19 and 2.20, we derive our final result for the case of pure translation:

$$E_t + \left(\frac{\mathbf{t} \cdot \hat{\mathbf{z}}}{\mathbf{R} \cdot \hat{\mathbf{z}}}\right) \Big(\mathbf{E_r} \cdot (\mathbf{r} - \mathbf{r}_0)\Big) = 0 \tag{2.21}$$

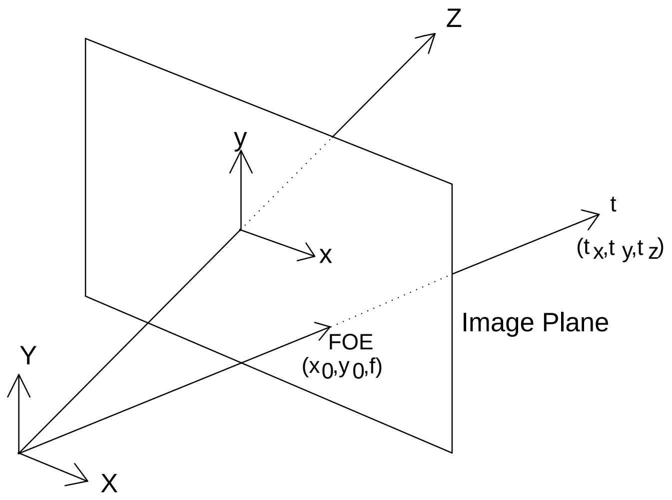


FIGURE 2-2: Definition of FOE location as intersection of velocity vector **t** with the image plane.

which is the *brightness change constraint equation* for the case of *pure translation*. This can also be expressed using components as

$$\boxed{\begin{array}{c} \tau E_t + (x - x_0)E_x + (y - y_0)E_y = 0 \\ \tau \equiv Z(x,y)/t_z \end{array}} \tag{2.22}$$

The *time to impact* τ is the ratio of the depth Z to the velocity along the optic axis t_z. 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.

2.2 Algorithms Based on the Brightness Change Constraint Equation

Having derived the brightness change contraint equation, we can now turn to algorithms utilizing this differential constraint with the goal of constructing a dedicated analog VLSI chip to estimate camera motion through the FOE .

2.2.1 Rotation and Translation When Depth is Known or Structured

Finding both rotation and translation when scene structure is also unknown is difficult chiefly due to the dependence of Z on x and y. For example, if we have $n \times n$ brightness gradient measurements, then we have n^2 constraints provided by the brightness change constraint equation but 6 more unknowns than that, producing an underconstrained system. If we assume that somehow we are provided with the depth map $Z(x,y)$, then the problem becomes overconstrained with six unknowns and n^2 constraints. Hence, we utilize least-square minimization to find a solution. We can minimize with respect to $\boldsymbol{\omega}$ and $\mathbf{t}$ an integral over the image of the squared error of the constraint equation:

$$\min_{\boldsymbol{\omega},\mathbf{t}} \iint_I \left(E_t + \frac{\mathbf{v}\cdot\boldsymbol{\omega}}{f} + \frac{\mathbf{s}\cdot\mathbf{t}}{Z} \right)^2 dx\,dy \tag{2.23}$$

Differentiating and setting the result equal to zero to find the extrema yields a closed form solution:

$$\begin{bmatrix} \iint_I \left(\frac{\mathbf{v}\mathbf{v}^T}{f^2}\right) dx\,dy & \iint_I \left(\frac{\mathbf{v}\mathbf{s}^T}{fZ}\right) dx\,dy \\ \iint_I \left(\frac{\mathbf{s}\mathbf{v}^T}{fZ}\right) dx\,dy & \iint_I \left(\frac{\mathbf{s}\mathbf{s}^T}{Z^2}\right) dx\,dy \end{bmatrix} \begin{bmatrix} \boldsymbol{\omega} \\ \mathbf{t} \end{bmatrix} = - \begin{bmatrix} \iint_I \left(\frac{E_t\mathbf{v}}{f}\right) dx\,dy \\ \iint_I \left(\frac{E_t\mathbf{s}}{Z}\right) dx\,dy \end{bmatrix} \tag{2.24}$$

Alternatively, if we know that the scene geometry is one of a parameterized family of surfaces, then we can also have an over-constrained system. For example, consider the simple case of a scene consisting of a single plane of unknown depth. The equation for a plane is:

$$\mathbf{R}\cdot\mathbf{n} = 1 \tag{2.25}$$

where $\mathbf{n}$ is a normal to the plane. Imposing the perspective projection equation (2.3) implies:

$$Z = \frac{f}{\mathbf{r}\cdot\mathbf{n}} \tag{2.26}$$

Combining this with the brightness change constraint equation, we have:

$$fE_t + \mathbf{v}\cdot\boldsymbol{\omega} + (\mathbf{r}\cdot\mathbf{n})(\mathbf{s}\cdot\mathbf{t}) = 0 \tag{2.27}$$

Now, we have nine unknowns (the components of $\mathbf{t}$, $\boldsymbol{\omega}$, and $\mathbf{n}$) and can solve using least squares methods. Again, we can minimize the sum of the squared error over the image:

$$\min_{\boldsymbol{\omega},\mathbf{t},\mathbf{n}} \iint_I \Big(fE_t + \mathbf{v}\cdot\boldsymbol{\omega} + (\mathbf{r}\cdot\mathbf{n})(\mathbf{s}\cdot\mathbf{t}) \Big)^2 dx\,dy \tag{2.28}$$

Differentiating this expression and setting the result equal to zero gives:

$$\begin{aligned} \iint_I \Big(fE_t + \mathbf{v}\cdot\boldsymbol{\omega} + (\mathbf{r}\cdot\mathbf{n})(\mathbf{s}\cdot\mathbf{t}) \Big)\mathbf{v}\,dx\,dy &= \mathbf{0} \\ \iint_I (\mathbf{r}\cdot\mathbf{n}) \Big(fE_t + \mathbf{v}\cdot\boldsymbol{\omega} + (\mathbf{r}\cdot\mathbf{n})(\mathbf{s}\cdot\mathbf{t}) \Big)\mathbf{s}\,dx\,dy &= \mathbf{0} \\ \iint_I (\mathbf{s}\cdot\mathbf{t}) \Big(fE_t + \mathbf{v}\cdot\boldsymbol{\omega} + (\mathbf{r}\cdot\mathbf{n})(\mathbf{s}\cdot\mathbf{t}) \Big)\mathbf{r}\,dx\,dy &= \mathbf{0} \end{aligned} \tag{2.29}$$

Even though these equations are nonlinear, there are iterative methods for their solution and, in fact, a closed form solution using eigenvalue/eigenvector analysis [26].

Another approach to simplifying the situation imposed by the local variation of the depth map Z is to remove the translational component entirely. When we only have rotation, then the brightness change constraint equation becomes especially simple:

$$fE_t + \mathbf{v}\cdot\boldsymbol{\omega} = 0 \tag{2.30}$$

This is always an over-constrained problem with three unknowns and n^2 constraints, one for every picture cell. We can minimize the square of the error of this equation:

$$\min_{\boldsymbol{\omega}} \iint_I \left(E_t + \frac{\mathbf{v} \cdot \boldsymbol{\omega}}{f} \right)^2 dx\,dy \tag{2.31}$$

to find the solution

$$\left[\iint_I \left(\mathbf{v}\mathbf{v}^T \right) dx\,dy \right] \boldsymbol{\omega} = -f \iint_I (\mathbf{v} E_t)\, dx\,dy \tag{2.32}$$

which is clearly just a special case of Equation 2.24.

To examine when this least-squares method for recovering rotational velocity becomes ill-conditioned, an analysis was performed in [21] assuming a uniform distribution of the allowed directions of $\mathbf{v}$ which demonstrated that the L^2 condition number of the matrix in the left hand side of Equation 2.32 increases as the extent of the *field of view* (FOV) decreases. The FOV is the cone of $\mathbf{r}$ directions defined by the principal distance f and finite extent of the image plane. With this analysis, the condition number reaches a minimum at a half-angle $\theta_v = 57.42°$. In general, the third component of $\boldsymbol{\omega}$, the rotation about the optic axis, is not recovered as accurately as the other two. The relative error between the third component and the other two varies inversely with the FOV [21].

Obviously, the more difficult situation due to the local variation of Z, and hence the more interesting one, is when we have some sort of translational motion. For simplicity, we will from now on eliminate the rotational component. Again, known scene geometry can greatly reduce the difficulty of the problem. As an example of a plain scene geometry, we can assume the presence of a plane perpendicular to the optic axis at a distance Z_0. In this case the solution becomes:

$$\left[\iint_I \left(\mathbf{s}\mathbf{s}^T \right) dx\,dy \right] \mathbf{t}' = - \iint_I (\mathbf{s} E_t)\, dx\,dy \tag{2.33}$$

$$\mathbf{t}' \equiv \frac{\mathbf{t}}{Z_0} \tag{2.34}$$

The third component of $\mathbf{t}'$ is the now constant time to impact τ. Notice the similarity to the solution for the case of pure rotation in Equation 2.32. The observations we noted for pure rotation also hold in this case. The condition number of the matrix in Equation 2.33 decreases with increasing FOV, reaching a minimum at half-angle $\theta_v = 54.74°$, and the accuracy with which we recover the third component is always less than the other two and varies inversely with the FOV. Thus, when examining rotation and translation separately, we find that a large FOV is required for accuracy. This is likely to also be necessary when recovering rotation and translation simultaneously because it becomes very difficult to differentiate between rotational and translational motion for small FOVs.

Since we are interested in situations where the scene geometry is completely unknown, we now restrict our attention to the special case of pure translation with arbitrary scene geometry.

2.2.2 Translation: The Eigenvalue/Eigenvector Solution

Even when we choose to eliminate rotation, it is difficult to solve for the translational motion once again due to the local variation of Z. There are three approaches that we have explored for estimation of the translational motion alone. Through introduction of the FOE, these problems can be restated to yield simpler solutions. It is these simplified solutions that appear promising for implementation in analog VLSI.

Image points at which $E_t = 0$ provide important constraints on the direction of translation; they are referred to as *stationary points* [21]. With $E_t = 0$, the brightness constraint equation becomes:

$$\frac{\mathbf{s} \cdot \mathbf{t}}{Z} = 0 \tag{2.35}$$

With finite and nonzero Z, this implies that $\mathbf{s}$ and $\mathbf{t}$ are orthogonal. Ideally, nonparallel $\mathbf{s}$ vectors at two stationary points are sufficient to calculate the direction of the translation vector $\mathbf{t}$, and hence the FOE. Of course, due to noise, the constraint equation will not be satisfied exactly, so we minimize the L^2 norm of the error of the equation for the set of stationary points $\{P_n\}$:

$$\min_{\mathbf{t},\lambda} \sum_{\{P_n\}} (\mathbf{s} \cdot \mathbf{t})^2 + \lambda(1 - \|\mathbf{t}\|^2) \tag{2.36}$$

Here we have added an additional constraint of $||\mathbf{t}|| = 1$ to account for the scale factor ambiguity. Clearly, we cannot estimate the absolute magnitude of the translation vector in the absence of additional information about Z. We can, however, estimate its direction. The resulting solution of this minimization is the eigenvalue/eigenvector problem:

$$\left[\sum_{\{P_n\}} \mathbf{s}\mathbf{s}^T \right] \mathbf{t} = \lambda \mathbf{t} \tag{2.37}$$

where we take as our solution the eigenvector corresponding to the smallest eigenvalue.

In order to understand more clearly the properties of this algorithm, we look more closely at the geometry of the brightness change constraint equation [21]. Knowing the permissible directions of $\mathbf{r}$ due to a finite field of view and noting that $\mathbf{s} \cdot \mathbf{r} = 0$ from Equation 2.13, allows us to calculate the region of allowed $\mathbf{s}$ directions. The directions of $\mathbf{r}$ lie in a cone defined by the half angle θ of the field of view. For a given $\mathbf{r}$, $\mathbf{s} \cdot \mathbf{r} = 0$ defines a plane in which $\mathbf{s}$ must lie. This plane cuts the unit sphere in a great circle and the collection of such great circles for all allowed $\mathbf{r}$ in the finite image plane forms the *permissible band* of $\mathbf{s}$ directions as shown in Figure 2-3. This band extends on the unit sphere the same angle θ as the field of view. Additionally, each point in the image plane constrains the translational vector $\mathbf{t}$ through Equation 2.17 to be in one hemisphere which is termed the *compatible hemisphere*. To see this we form the $\bar{\mathbf{s}}$-projection [21]:

$$\bar{\mathbf{s}} = -\text{sign}(E_t)\mathbf{s} \tag{2.38}$$

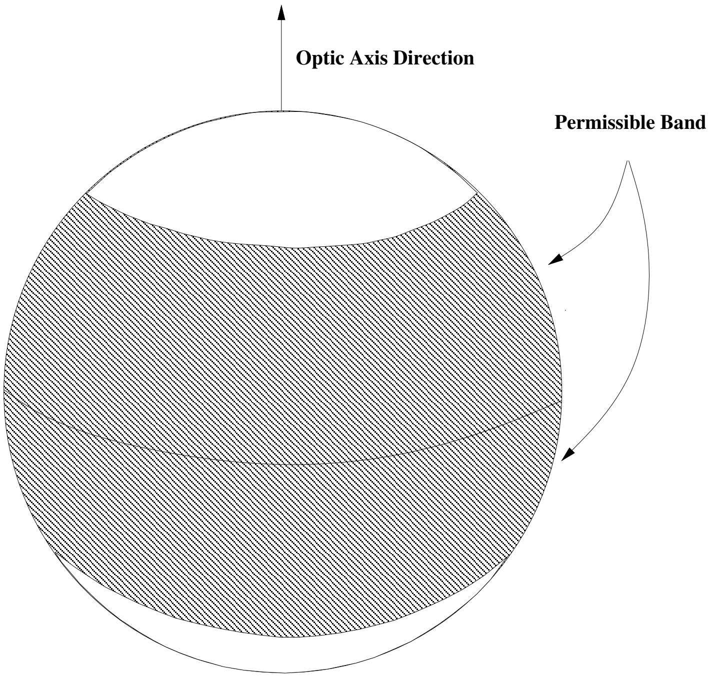


FIGURE 2-3: Permissible bands of $\mathbf{s}$ on the unit sphere. After [21].

We know that from Equation 2.17

$$E_t = -\frac{1}{Z}(\mathbf{s}\cdot\mathbf{t})$$
$$\Rightarrow \bar{\mathbf{s}}\cdot\mathbf{t} = \operatorname{sign}\left(\frac{1}{Z}\right)|\mathbf{s}\cdot\mathbf{t}| \tag{2.39}$$

Since imaged depth is positive, i.e. $Z > 0$, this implies:

$$\bar{\mathbf{s}}\cdot\mathbf{t} \geq 0 \tag{2.40}$$

Clearly, $\bar{\mathbf{s}}$ also lies in the same permissible band as $\mathbf{s}$, and hence $\mathbf{t}$ must be within 90° of $\bar{\mathbf{s}}$, defining a compatible hemisphere. For each pixel in our imaging array, we can calculate an $\bar{\mathbf{s}}$ and the intersection of all of the resultant compatible hemispheres forms a polygon in which the direction of the translation vector $\mathbf{t}$ must lie. This polygon has relatively few sides; they are formed by points where $\bar{\mathbf{s}}\cdot\mathbf{t} = 0$, i.e., constraints provided by stationary points. Thus, the algorithm described earlier essentially locates the best center of this region in a least squares sense. Equivalently, we can view the algorithm as finding the best great circle which fits the distribution of directions of the $\mathbf{s}$ vector for all stationary points as shown in Figure 2-4.

2.2.3 Translation: The Linear Solution

If we pose the problem in terms of the FOE (x_0, y_0), we can instead use the simpler form of Equation 2.21 and minimize with $t_z = f$:

$$\min_{\mathbf{r}_0} \sum_{\{P_n\}} \Big(\mathbf{E_r}\cdot(\mathbf{r}-\mathbf{r}_0)\Big)^2 \tag{2.41}$$

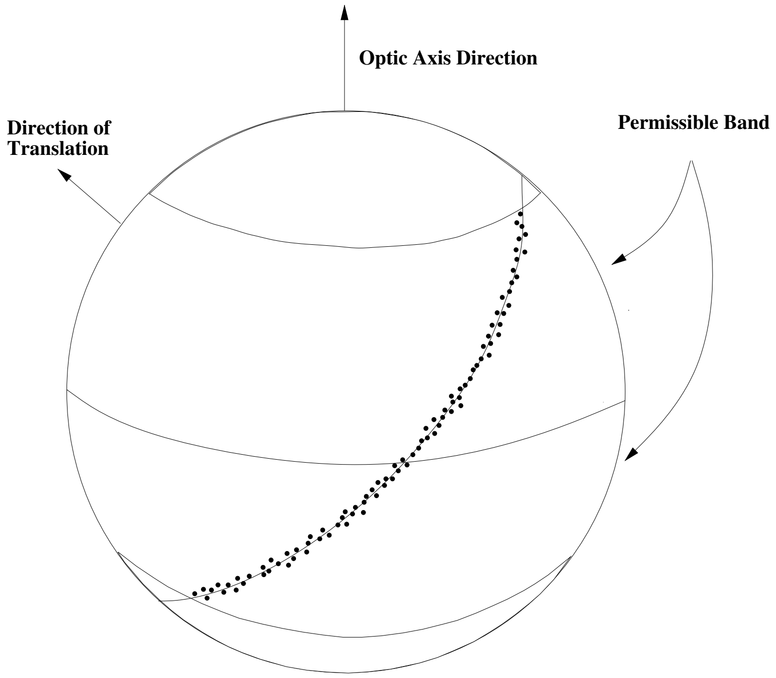


FIGURE 2-4: The great circle which fits the distribution of directions of $\mathbf{s}$ vector for points where $E_t \approx 0$. After [21].

A linear closed form solution for this slightly different problem is:

$$\left[\sum_{\{P_n\}} \left(\mathbf{E_r}\mathbf{E_r}^T\right)\right] \mathbf{r}_0 = \sum_{\{P_n\}} \left(\mathbf{E_r}\mathbf{E_r}^T\right) \mathbf{r} \tag{2.42}$$

Note that this equation involves no dependence on the principle distance f and is independent of the coordinate system origin! We do not need to know f or the location of the principal point to find the FOE. We do need to know the camera calibration in the case of Equation 2.37 when we were solving for the translation vector $\mathbf{t}$.

In general, the approach of posing the problem in terms of the FOE and minimizing with $t_z = f$ leads to a linear solution. This is usually the simplest answer, and therefore the most appealing for realization in analog VLSI.

2.2.4 Translation: Minimizing Z^2

It is clear that good constraints of the FOE occur at points where $E_t = 0$, usually a small set. This indicates that our answer is based on a few critical points and is therefore not especially robust. A modification of the preceding algorithm provides a means of estimating the FOE without explicitly finding this set and extends the computation to the entire image [21]. If we use an assumed velocity vector $\mathbf{t}'$ as an estimate for $\mathbf{t}$ in Equation 2.17, we get incorrect depth values:

$$Z' = Z\left(\frac{\mathbf{s}\cdot\mathbf{t}'}{\mathbf{s}\cdot\mathbf{t}}\right) \tag{2.43}$$

This equation implies that we can get negative depth values for incorrect choice of $\mathbf{t}'$. Additionally, we can get values that will be very large, both positive and negative, near stationary points (where $\mathbf{s}\cdot\mathbf{t}\approx 0$). This suggests that, assuming the range of Z is finite, minimization of the integral of Z^2 subject to $\|\mathbf{t}\|^2 = 1$ may be a good method to find the translation vector:

$$\min_{\mathbf{t},\lambda}\left\{\iint_I \frac{(\mathbf{s}\cdot\mathbf{t})^2}{E_t^2+\eta^2}dxdy + \lambda\left(1-\|\mathbf{t}\|^2\right)\right\} \tag{2.44}$$

where, in order to reduce potential problems with noise where $E_t \approx 0$, a positive constant has been added to E_t^2. The solution to this minimization problem is once again an eigenvalue/eigenvector problem:

$$\left[\iint_I \left(\frac{\mathbf{s}\mathbf{s}^T}{E_t^2+\eta^2}\right)dxdy\right]\mathbf{t} = \lambda\mathbf{t} \tag{2.45}$$

where we take as our solution the eigenvector corresponding to the smallest eigenvalue. Posing our problem in terms of the FOE with $t_z = f$ gives instead the simpler problem:

$$\min_{\mathbf{r}_0}\iint_I \frac{\left(\mathbf{E_r}\cdot(\mathbf{r}-\mathbf{r}_0)\right)^2}{E_t^2+\eta^2}dxdy \tag{2.46}$$

It is important to note that for points where $E_t = 0$, this error function is essentially the same as in Equation 2.41. Points other than stationary points are then weighted inversely with E_t^2 using a Cauchy function:

$$W(E_t,\eta) = \frac{1}{E_t^2+\eta^2} \tag{2.47}$$

Now the integrals in this equation are over the whole image, as opposed to sums over the set of stationary points as in Equation 2.41. The solution to this problem becomes:

$$\left[\iint_I W(E_t,\eta)\left(\mathbf{E_r}\mathbf{E_r}^T\right)dxdy\right]\mathbf{r}_0 = \iint_I W(E_t,\eta)\left(\mathbf{E_r}\mathbf{E_r}^T\right)\mathbf{r}dxdy \tag{2.48}$$

or equivalently:

$$\begin{aligned}\iint_I W(E_t,\eta)E_x\Big((x-x_0)E_x+(y-y_0)E_y\Big)dxdy &= 0\\ \iint_I W(E_t,\eta)E_y\Big((x-x_0)E_x+(y-y_0)E_y\Big)dxdy &= 0\end{aligned} \tag{2.49}$$

The actual functional form of the weighting with E_t in Equation 2.47 may not be essential, as long as the weight is small for large E_t and large for small E_t. Thus, we may be able to use something much simpler, perhaps even a cutoff above $|E_t| > \eta$ [27, 23]. The new cutoff weighting function becomes:

$$W(E_t,\eta) = \begin{cases} 1 & \text{if } |E_t| < \eta \\ 0 & \text{otherwise}\end{cases} \tag{2.50}$$

2.2.5 Translation: Minimizing the Number of Negative Depth Values

Another promising approach to finding the FOE is based on the *depth-is-positive constraint* [28, 29, 30]. As noted in Equation 2.43, incorrect choice of vector $\mathbf{t}'$ results in negative $Z'(x, y)$. Imaged depth, however, must be positive.

We want methods which attempt to find the $\mathbf{t}'$ which minimizes the number of negative Z' values across the image:

$$\min_{x_0, y_0} \iint_I \mathrm{u}\left(E_t \Big((x - x_0) E_x + (y - y_0) E_y \Big) \right) dx dy. \tag{2.51}$$

where $\mathrm{u}(t)$ is the unit step function. Because the functional is not convex, this is a difficult problem to solve. We can attempt to find the minimum by searching over the available solution space. For each choice of (x_0, y_0) we can calculate the total number of negative depth values. We can choose as an initial estimate of the FOE the location on a coarse tessellation of the image plane which has a minimum number negative depth values. Then, to achieve better resolution, we can perform finer searches about this estimate, until we reach the desired accuracy. Of course, this is highly computationally intensive and inefficient. To avoid this, we can formulate algorithms which use a convex error function instead.

One idea attempts to utilize a convex error functional which penalizes negative Z' much more heavily than positive Z' such as:

$$\min_{x_0, y_0} \iint_I \exp\left(\beta E_t \left((x - x_0) E_x + (y - y_0) E_y \right) \right) dx dy \tag{2.52}$$

where β controls the steepness of the penalty for negative Z'. Alternatively, we can formulate a convex error functional which penalizes negative depth values only:

$$\min_{x_0, y_0} \iint_I \Big(-E_t^2 \tau(x_0, y_0) \Big)^{\alpha} \mathrm{u}\Big(-\tau(x_0, y_0) \Big) dx dy \tag{2.53}$$

Choosing $\alpha > 1$ makes each term in the sum strictly convex, and hence the sum is also strictly convex. We can additionally weight the terms in this sum by our confidence in the robustness of the measurements with respect to noise. Using a confidence measure that is proportional to both E_t and the magnitude of $\mathbf{E_r}$, we have

$$\min_{\mathbf{r}_0} \iint_I ||E_t \mathbf{E_r}||^p \Big(E_t \mathbf{E_r} \cdot (\mathbf{r} - \mathbf{r}_0) \Big)^{\alpha} \mathrm{u}\Big(E_t \mathbf{E_r} \cdot (\mathbf{r} - \mathbf{r}_0) \Big) dx dy \tag{2.54}$$

where p controls our confidence in the data. This method should be less noise sensitive when compared with the other methods that we have discussed.

Yet another idea attempts to mesh the Z^2 approach with the idea embodied in Equation 2.51, by using a bi-quadratic error function:

$$\min_{\mathbf{r}_0} \iint_I \left(\frac{\Big(\mathbf{E_r} \cdot (\mathbf{r} - \mathbf{r}_0) \Big)^2}{E_t^2 + \eta^2} \right) \Big[\Lambda \mathrm{u}\Big(E_t \mathbf{E_r} \cdot (\mathbf{r} - \mathbf{r}_0) \Big) + \mathrm{u}\Big(- E_t \mathbf{E_r} \cdot (\mathbf{r} - \mathbf{r}_0) \Big) \Big] dx dy \tag{2.55}$$

where Λ controls the penalty assigned negative depth values compared to positive depth values. The solution to this problem then becomes:

$$\left[\iint_I W\frac{\left(\mathbf{E_r}\mathbf{E_r}^T\right)}{E_t^2+\eta^2}dxdy\right]\mathbf{r}_0 = \iint_I W\frac{\left(\mathbf{E_r}\mathbf{E_r}^T\right)\mathbf{r}}{E_t^2+\eta^2}dxdy \tag{2.56}$$

$$W = \Lambda \mathrm{u}\Big(E_t\mathbf{E_r}\cdot(\mathbf{r}-\mathbf{r}_0)\Big) + \mathrm{u}\Big(-E_t\mathbf{E_r}\cdot(\mathbf{r}-\mathbf{r}_0)\Big) \tag{2.57}$$

For a given Λ, these equations can be solved iteratively. If we set $\Lambda = 1$, we are solving the Z^2 problem. By continuing on Λ, we penalize negative depth values more heavily. Of course, we can use alternative weighting with E_t such as the cutoff weighting function in lieu of the Cauchy weight we have shown here.

2.2.6 Translation: Algorithms Based on the Distribution of Negative Depth Values

All of the algorithms based on the depth-is-positive constraint that we have discussed so far have been complex or inefficient, and therefore not suitable for implementation. Perhaps a more promising algorithm is based on the distribution of negative depth values for a given conjectured location of the FOE (x_0', y_0') [30]. We require that

$$ZZ' = \left(\frac{t_z}{E_t}\right)^2\Big(\mathbf{E_r}\cdot(\mathbf{r}-\mathbf{r}_0)\Big)\Big(\mathbf{E_r}\cdot(\mathbf{r}-\mathbf{r}_0')\Big) > 0 \tag{2.58}$$

since $Z > 0$ and a possible solution Z' must also satisfy $Z' > 0$. If t_z/E_t is finite and nonzero, this effectively reduces to

$$(\mathbf{E_r}\cdot\mathbf{x})(\mathbf{E_r}\cdot\mathbf{x}') > 0 \tag{2.59}$$

where $\mathbf{x}$ is the vector from the point $\mathbf{r}$ to the true location of the FOE and $\mathbf{x}'$ is the vector from the point $\mathbf{r}$ to the assumed location of the FOE. For a given point $\mathbf{r}$, this equation divides the possible directions of $\mathbf{E_r}$ into a permissible and forbidden region. Examining Figure 2-5, we draw the vectors $\mathbf{x}$ and $\mathbf{x}'$ for a given point P. For each vector, we then draw the line perpendicular to that vector. This divides the plane into four regions. If $\mathbf{E_r}$ lies in the permissible regions, where the projection on both vectors is of the same sign, then the constraint $ZZ' > 0$ is satisfied. In the forbidden regions, the projections are of opposite sign, and the constraint is violated. Since we assume that we know nothing a priori about the image, we can use a model which assigns a uniform probability distribution to the possible directions of $\mathbf{E_r}$. In this case, the angle subtended by the forbidden region is a measure of the probability that the depth-is-positive constraint is violated at that point. Proceeding in this manner for every point P in the image plane, we can define a probability map as shown in Figure 2-6. For example, for points P on the line segment connecting the true FOE to the conjecture, the constraint will be violated with probability 1. Additionally, on the extension of the line segment past the two foci, the probability of a negative depth value is zero. Elsewhere, the map indicates that points where

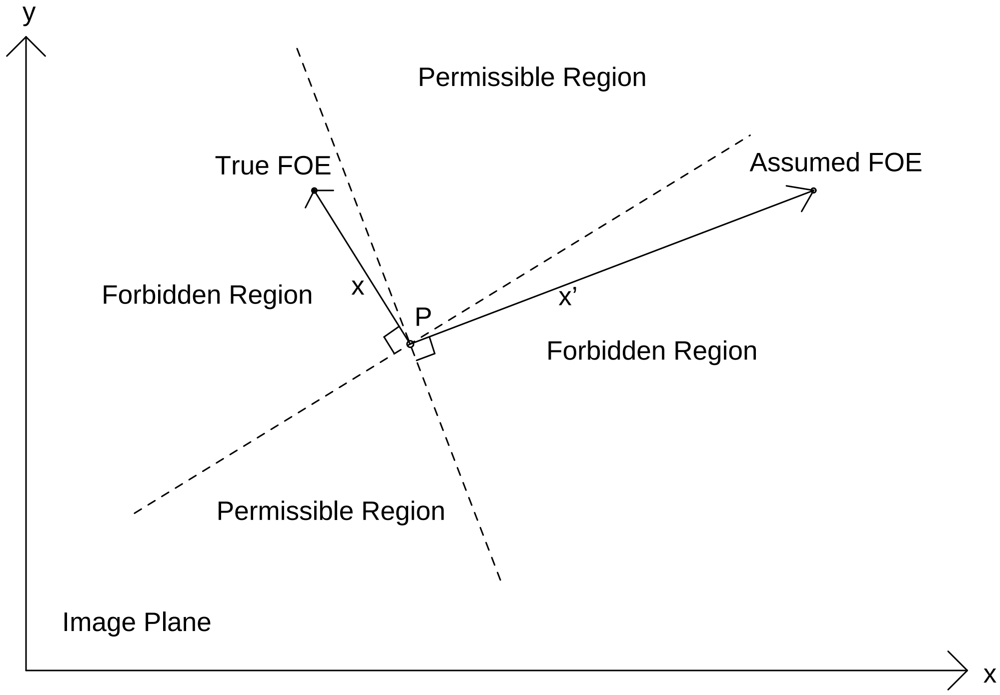


FIGURE 2-5: Determination of permissible and forbidden regions for directions of $\mathbf{E_r}$.

the constraint is violated will tend to cluster about this line segment, with the probability of a negative depth value dropping inversely with the perpendicular distance from the line. These observations lead to a variety of algorithmic ideas.

We can assume an estimate for the FOE, find the negative depth bit map using

$$\mathrm{sign}(Z') = \mathrm{u}\left(E_t\Big((x_0 - x)E_x + (y_0 - y)E_y\Big)\right) \tag{2.60}$$

and locate the "constraint" line which connects the conjecture to the true FOE. Ideally, the intersection of two different constraint lines will give a good estimate of the true FOE. In the presence of noise in the image brightness, we would want to find more than two constraint lines, and then calculate the true FOE as the point which minimizes the sum of squared distances from these constraint lines [30]. This is rather complicated; a simpler method would be to assume a conjecture, find the centroid of the resulting negative depth bit map, use this as the new estimate of the location of the true FOE, and iterate as necessary. The centroid $\mathbf{r}_0$ of a bit map is defined as:

$$\left[\iint_I b(x,y)\,dx\,dy\right]\mathbf{r}_0 = \iint_I b(x,y)\,\mathbf{r}\,dx\,dy \tag{2.61}$$

where $b(x,y)$ is our bit map. This is a reasonable calculation because the four-fold symmetry of the probability map implies that the expected location of the centroid lies halfway along the line connecting the two foci. This procedure can be thought of as "walking" down a constraint line.

However, there are two difficulties with this approach. First, if we assume that noise causes

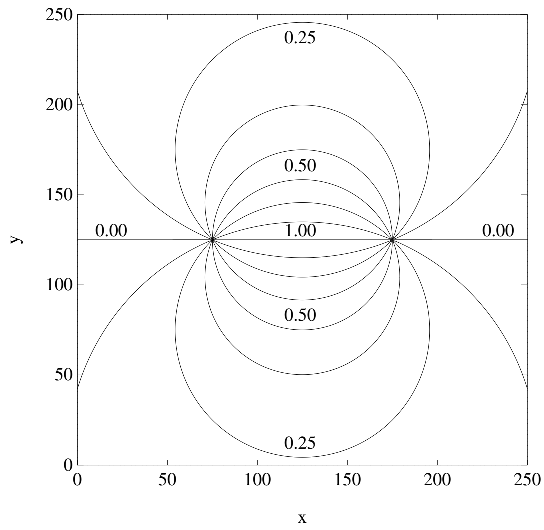


FIGURE 2-6: The map of probabilities that the depth-is-positive constraint is violated for a given conjecture. The two foci refer to the locations of the true and assumed location of the FOE.

a uniform distribution of negative depth values *even when we have the correct answer*, then the centroid will be biased away from the desired location and towards the center of the image. Secondly, the finite extent of the image plane can violate our assumption of symmetry, also causing the centroid to deviate. To alleviate these problems, we can restrict the centroid calculation to a region of support (ROS) around the current estimate of the FOE. In this case, we are essentially performing the following iteration:

$$\iint_I W\left(\mathbf{r}, \mathbf{r}_0^{(n-1)}\right) \mathrm{u}\left(E_t \mathbf{E_r} \cdot \left(\mathbf{r}_0^{(n-1)} - \mathbf{r}\right)\right) \left(\mathbf{r}_0^{(n)} - \mathbf{r}\right) dx dy = 0 \tag{2.62}$$

where $\mathbf{r}_0^{(n)}$ is the estimate of the FOE at the nth iteration and W is some weighting function which selects the ROS about the estimate of the FOE for the calculation. In practice, a weighting function of the form

$$W(\mathbf{r}, \mathbf{r}_0) = \begin{cases} 1 & \text{if } |x - x_0| < \lambda \text{ and } |y - y_0| < \lambda \\ 0 & \text{otherwise} \end{cases} \tag{2.63}$$

would probably be simple to implement.

2.2.7 Translation: Solutions Which Favor Smoothness of Depth

Ideally, we would like to minimize the following error functional:

$$\min_{\mathbf{t},Z} = \iint_I \Big(ZE_t + (\mathbf{s}\cdot\mathbf{t})\Big)^2 dxdy \tag{2.64}$$

Obviously, if the translation vector $\mathbf{t}$ is known, then we can set the integrand to zero at each point (x, y):

$$Z(x,y) = -\frac{(\mathbf{s}\cdot\mathbf{t})}{E_t} \tag{2.65}$$

When we do not know the motion parameters as well, we must provide an additional constraint to find a solution. A constraint we may apply is smoothness of the depth map $Z(x, y)$ or, equivalently, the time to impact map $\tau(x, y)$.

$$\min_{\mathbf{r}_0,\tau} \iint_I \left(\Big(\tau E_t + \mathbf{E_r}\cdot(\mathbf{r}-\mathbf{r}_0)\Big)^2 + \sigma(\tau_x^2+\tau_y^2)\right) dxdy \tag{2.66}$$

This is a 2-D calculus of variations problem. Given an error function of the form:

$$J = \iint_I F(x,y,\Psi,\Psi_x,\Psi_y)dxdy \tag{2.67}$$

we have the *Euler equation* which defines the functional solution $\Psi(x, y)$ which minimizes J [25]:

$$F_\Psi = \frac{\partial}{\partial x}F_{\Psi_x} + \frac{\partial}{\partial y}F_{\Psi_y} \tag{2.68}$$

with natural boundary conditions:

$$F_{\Psi_x}\frac{dy}{ds} = F_{\Psi_y}\frac{dx}{ds} \tag{2.69}$$

where s is a parameter which varies along the boundary of the region I [25]. For this problem, the Euler equation becomes:

$$\sigma\nabla^2\tau = \Big(\tau E_t + \mathbf{E_r}\cdot(\mathbf{r}-\mathbf{r}_0)\Big)E_t \tag{2.70}$$

$$\tau_x\frac{dy}{ds} - \tau_y\frac{dx}{ds} = 0 \tag{2.71}$$

To find the other parameters, we can at the same time minimize the functional with $t_z = f$:

$$\left[\iint_I \mathbf{E_r}\mathbf{E_r}^T dxdy\right]\mathbf{r}_0 = \iint_I \mathbf{E_r}\Big(\tau E_t + \mathbf{E_r}^T\mathbf{r}\Big)dxdy \tag{2.72}$$

Implementation of Euler equations of this Poisson-like form can be accomplished using two-dimensional resistive sheets in the continuous case and resistive grids in the discrete case [27]. Discontinuities in the depth map can also be accommodated perhaps through the use of fused resistors[3, 4, 31, 32].

2.3 Algorithmic Simulations

2.3.1 Synthetic Image Generation

In [23], the most promising of the proposed algorithms were extensively simulated in order to compare their accuracy, robustness, noise immunity, as well as their suitability for implementation in analog VLSI. For that investigation, synthetic image data of a scene consisting of a single arbitrarily oriented textured plane was exclusively used. The texture on the plane was a sinusoidal grating pattern of the form:

$$E_s(\alpha_1, \alpha_2) = \frac{E_0}{2}\left(1 + \sin\left(\frac{\alpha_1}{\lambda_1}\right)\sin\left(\frac{\alpha_2}{\lambda_2}\right)\right) \tag{2.73}$$

where α_1 and α_2 are coordinates in the object plane. This kind of pattern was used because of the large number of stationary points it provides during motion, and also because of the smoothly varying gradients of the resultant image sequences. The origin of this coordinate system is at $\mathbf{R}_0 = (X_0, Y_0, Z_0)^T$ in the camera coordinate system. Defining the vectors $\mathbf{x} = (x_1, x_2, x_3)^T$ and $\mathbf{y} = (y_1, y_2, y_3)^T$ as parallel to the axes in the object plane, we can define points in the plane as:

$$\mathbf{R} = \mathbf{R}_0 + \alpha_1\mathbf{x} + \alpha_2\mathbf{y} \tag{2.74}$$

Next using the perspective projection equation to find the 2-D mapping from plane coordinates (α_1, α_2) to image coordinates (x, y), we find:

$$\begin{bmatrix} fx_1 - xx_3 & fy_1 - xy_3 \\ fx_2 - yx_3 & fy_2 - yy_3 \end{bmatrix}\begin{bmatrix} \alpha_1(x,y) \\ \alpha_2(x,y) \end{bmatrix} = \begin{bmatrix} xZ_0 - fX_0 \\ yZ_0 - fY_0 \end{bmatrix} \tag{2.75}$$

To determine the correct vectors $\mathbf{x}$ and $\mathbf{y}$, we take the coordinate system in the image plane, rotate it by θ_1 about the x axis and then rotate it by θ_2 about the y axis. We perform these rotations using orthonormal rotation matrices resulting in:

$$\mathbf{x} = \begin{bmatrix} \cos\theta_2 \\ 0 \\ \sin\theta_2 \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} \sin\theta_1\sin\theta_2 \\ \cos\theta_1 \\ -\sin\theta_1\cos\theta_2 \end{bmatrix} \tag{2.76}$$

In a constant velocity motion sequence we merely change the location of the origin of the coordinate system of the object plane

$$\mathbf{R}_0(t) = \Big(X_0(t), Y_0(t), Z_0(t)\Big)^T \tag{2.77}$$

at a constant rate $\mathbf{t} = (t_x, t_y, t_z)^T$. The FOE set up by this motion is:

$$\begin{bmatrix} x_0 \\ y_0 \end{bmatrix} = \begin{bmatrix} f\left(\frac{t_x}{t_z}\right) \\ f\left(\frac{t_y}{t_z}\right) \end{bmatrix} \tag{2.78}$$

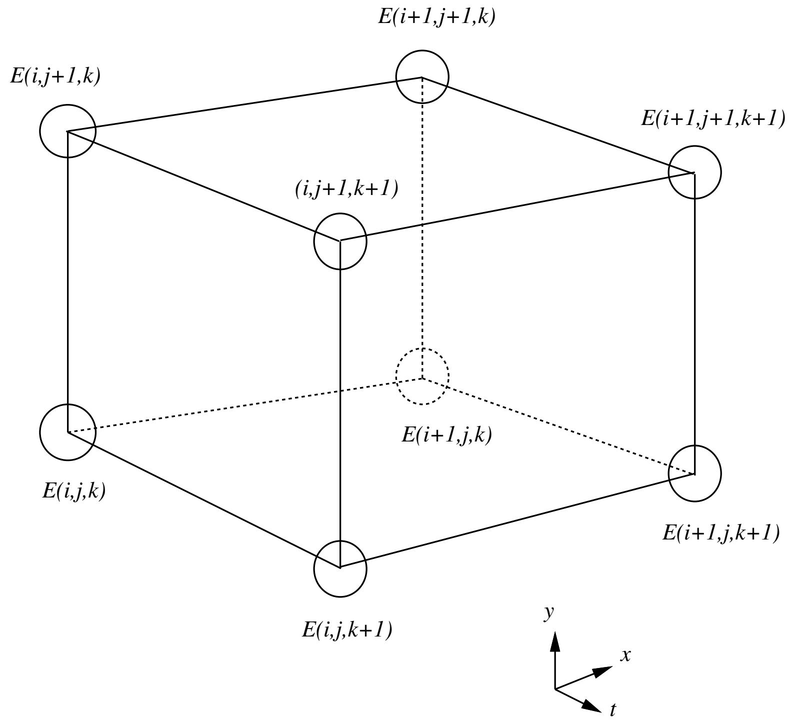


FIGURE 2-7: The three partial derivatives of image brightness at the center of the cube are estimated from the average of the first four differences along the four parallel edges. Here i is in the x direction, j is in the y direction and k is in the time direction.

In order to approximate the brightness gradients E_x, E_y, and E_t, we require a pair of images taken sequentially in time. From this image pair, we can use centered finite differencing to estimate the gradients. It can be shown [18] that the first order difference approximations to these partial derivatives with the lowest order error terms in their Taylor expansions are:

$$\begin{aligned} E_x \approx \frac{1}{4}\Big(&\big(E(i,j+1,k)-E(i,j,k)\big)+\big(E(i+1,j+1,k)-E(i+1,j,k)\big) \\ &+\big(E(i,j+1,k+1)-E(i,j,k+1)\big)+\big(E(i+1,j+1,k+1)-E(i+1,j,k+1)\big)\Big) \end{aligned} \tag{2.79}$$

$$\begin{aligned} E_y \approx \frac{1}{4}\Big(&\big(E(i+1,j,k)-E(i,j,k)\big)+\big(E(i+1,j+1,k)-E(i,j+1,k)\big) \\ &+\big(E(i+1,j,k+1)-E(i,j,k+1)\big)+\big(E(i+1,j+1,k+1)-E(i,j+1,k+1)\big)\Big) \end{aligned} \tag{2.80}$$

$$\begin{aligned} E_t \approx \frac{1}{4}\Big(&\big(E(i,j,k+1)-E(i,j,k)\big)+\big(E(i+1,j,k+1)-E(i+1,j,k)\big) \\ &+\big(E(i,j+1,k+1)-E(i,j+1,k)\big)+\big(E(i+1,j+1,k+1)-E(i+1,j+1,k)\big)\Big) \end{aligned} \tag{2.81}$$

where we refer to the cube of pixels in Figure 2-7. Clearly, each of our estimates is the average of the first four differences taken over adjacent measurements in this cube. As such, the estimate applies best at the center of the cube.

2.3.2 Simulation Results

In [23], we examined the following algorithms as potential candidates for implementation:

- The eigenvector/eigenvalue solution, both with the Cauchy weighting of Equation 2.47 and with the cutoff weighting of Equation 2.50.
- The linear solution also with Cauchy weighting as in Equation 2.47 and cutoff weighting as in Equation 2.48.
- The one-sided convex error function penalizing negative depth values as in Equation 2.54.
- A least squares intersection of constraint lines based on the distribution of negative depth values.
- An iterative method based on "walking" down constraint lines given by the distribution of negative depth values.

The methods based on the distribution of negative depth values did not perform as well as the other methods, and hence were eliminated from consideration. The one sided convex error function, a method based on the number of negative depth values, did have better performance than the cutoff linear solution, but the increased complexity (estimated to be $> 20\%$ based on implementation considerations) more than made up for the enhanced performance. In comparing the eigensolution to the linear solution, we reached the following conclusions:

- The simpler linear solution has acceptable absolute error, and is marginally poorer in performance in general when compared with the full-blown eigenvalue/eigenvector solution. The simplicity of the linear solution is an acceptable trade-off for the degraded performance.
- All solution methods display a bias for nonzero η which gets worse the farther away from the optic axis the FOE is. This error can be kept manageable by having a small η in the weighting function. However, when the FOE starts to leave the field of view the error gets much worse. This is understandable strictly from the geometry of the information garnished from the image data in such a situation. The $\mathbf{s}$ directions must all fall into the permissible band, which depends on the FOV of the camera. Consider the situation when we intersect a narrow band (whose width is controlled by η in our algorithms) of $\mathbf{s}$ locations about the great circle on which the stationary points lie with the permissible band. If the angle between the optic axis and the translational vector is small, then this narrow band falls entirely within the permissible band. Thus, their intersection goes all the way around the sphere, symmetrical about a plane splitting it in half, the normal of which is the desired translational vector. In this case, we expect small error with increasing η. However, if the angle is made large enough, then the circular band defined

by the translation vector exits from the permissible band, and hence is truncated. This happens when the FOE is moved near the edge of the FOV and continues to be the case when it moves outside. The resulting regions are basically two parallelogram-like shapes on opposite sides of the sphere which become progressively smaller as **t** is rotated outside of the field of view. This sectioning of the data drastically reduces the sensitivity of the sum we are minimizing to errors in the direction of **t** and consequently the error in the solution for the FOE found by our algorithms is quite large. Hence, we can only hope to find the FOE with reasonable accuracy in a region about the principal point and ending away from the image edges.

- The actual form of the weighting function in the linear solution is unimportant, as long as it is large for small E_t and small for large E_t. Hence, we can use the simpler cutoff weighting function.

- The solution methods appear robust with respect to degradation due to noise. The noise added in the simulation can be thought of as modeling real noise, random offsets and nonlinearities that inevitably arise in the circuitry that can be used to implement the algorithm.

Hence, the linear solution with cutoff weighting appeared to be a viable candidate for implementation from an algorithmic point of view. Additionally, it is also possible to implement this algorithm in an analog VLSI chip, as we shall see in Chapter 3.

3

The FOE Chip System Architecture and Circuit Design

3.1 Potential Approaches

Now we turn to potential implementations of the algorithm we have chosen. The system we design will solve:

$$\min_{\mathbf{r}_0} \iint_I W(E_t, \eta)\Big(\mathbf{E_r} \cdot (\mathbf{r} - \mathbf{r}_0)\Big)^2 dxdy \tag{3.1}$$

$$W(E_t, \eta) = \begin{cases} 1 & \text{if } |E_t| < \eta \\ 0 & \text{otherwise} \end{cases} \tag{3.2}$$

where the cutoff weighting function $W(E_t, \eta)$ selects those points in the image I which are stationary points. The closed form linear solution to this problem is:

$$\left[\iint_I W(E_t, \eta)\left(\mathbf{E_r}\mathbf{E_r}^T\right) dxdy\right] \mathbf{r}_0 = \iint_I W(E_t, \eta)\left(\mathbf{E_r}\mathbf{E_r}^T\right) \mathbf{r} dxdy \tag{3.3}$$

or equivalently:

$$\begin{bmatrix} \iint_I W(E_t, \eta) E_x^2 dxdy & \iint_I W(E_t, \eta) E_x E_y dxdy \\ \iint_I W(E_t, \eta) E_x E_y dxdy & \iint_I W(E_t, \eta) E_y^2 dxdy \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} = - \begin{bmatrix} \iint_I W(E_t, \eta) E_x (xE_x + yE_y)\, dxdy \\ \iint_I W(E_t, \eta) E_y (xE_x + yE_y)\, dxdy \end{bmatrix}$$

We could design a system to calculate the elements of this matrix equation and solve for the FOE off-chip. This would require the on-chip calculation of five complex quantities. The complexity of these quantities makes such an approach prohibitive. We instead would like a system to calculate the two components of the location of the FOE using gradient descent. By using such a feedback 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 $\alpha = (\alpha_0, \cdots, \alpha_{N-1})^T$ with a minimum, we can minimize this function via:

$$\frac{d\alpha}{dt} = -\beta \nabla_\alpha f(\alpha) \tag{3.4}$$

where β is a positive definite matrix. If we use the L^2 norm for f, the function is convex and a global minimum can exist. Then we can use gradient descent to solve Equation 3.1 to find:

$$\begin{aligned}
\frac{dx_0}{dt} &= \beta_0 \iint_I W(E_t, \eta) E_x \Big((x - x_0) E_x + (y - y_0) E_y \Big) dx dy \\
\frac{dy_0}{dt} &= \beta_1 \iint_I W(E_t, \eta) E_y \Big((x - x_0) E_x + (y - y_0) E_y \Big) dx dy
\end{aligned}$$

We could follow the lead of Tanner and Mead [2, 33] and build a massively parallel analog system to implement this. We would use photo-transistors as our imaging devices and estimate the brightness gradient in time using differentiators, and the brightness gradient in space using finite differences. Discretizing Equation 2.49, we arrive at:

$$\begin{aligned}
\frac{dx_0}{dt} &= \beta_0 \sum_{(x,y)\in I} W(E_t, \eta) E_x \Big((x - x_0) E_x + (y - y_0) E_y \Big) \\
\frac{dy_0}{dt} &= \beta_1 \sum_{(x,y)\in I} W(E_t, \eta) E_y \Big((x - x_0) E_x + (y - y_0) E_y \Big) \qquad (3.5)
\end{aligned}$$

We can build an $n \times n$ array of analog processors coupled with photo-sensors. A processor at position (x, y) in this array would calculate two currents:

$$\begin{aligned}
\Delta I_x &\propto W(E_t, \eta) E_x \Big((x - x_0) E_x + (y - y_0) E_y \Big) \qquad (3.6) \\
\Delta I_y &\propto W(E_t, \eta) E_y \Big((x - x_0) E_x + (y - y_0) E_y \Big)
\end{aligned}$$

These currents are injected by each processor into global busses for the voltages x_0 and y_0 respectively, thereby accomplishing the summations by Kirchoff's current law. The total currents injected into the busses would be:

$$\begin{aligned}
(\Delta I_x)_{\text{total}} &\propto \sum_{(x,y)\in I} W(E_t, \eta) E_x \Big((x - x_0) E_x + (y - y_0) E_y \Big) \qquad (3.7) \\
(\Delta I_y)_{\text{total}} &\propto \sum_{(x,y)\in I} W(E_t, \eta) E_y \Big((x - x_0) E_x + (y - y_0) E_y \Big)
\end{aligned}$$

These busses are terminated with capacitances C_x and C_y. The capacitors satisfy:

$$\begin{aligned}
\frac{dx_0}{dt} &= \frac{(\Delta I_x)_{\text{total}}}{C_x} \\
\frac{dy_0}{dt} &= \frac{(\Delta I_y)_{\text{total}}}{C_y} \qquad (3.8)
\end{aligned}$$

implementing Equation 3.5.

The major difficulty with this approach is that of area. The currents that the processors calculate require four multiplies and the cutoff weighting function. Including all of this circuitry per pixel in addition to the photo-transistors creates a very large pixel area. The number of pixels that we would be able to put on a single chip would therefore be small. The actual number of pixels contributing to our computation is small to begin with because the number of stationary points in the image is only a fraction of the total number of pixels. Thus, a large number of pixels is 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, with the rest idle.

3.2 Richardson's Iterative Method

To increase the number of pixels and make more efficient use of area, the solution that was decided upon was to multiplex the system. Instead of computing all of the terms of our summation in parallel, we calculate them sequentially. Of course, we can no longer use the derivatives in the right hand side of Equation 3.5. We can use a forward difference approximation to model the gradient descent equation. Our new system of equations would become:

$$\begin{aligned} x_0^{(i+1)} &= x_0^{(i)} + h \sum_{(x,y)\in I} W(E_t,\eta)E_x \left(\left(x - x_0^{(i)}\right) E_x + \left(y - y_0^{(i)}\right) E_y \right) \\ y_0^{(i+1)} &= y_0^{(i)} + h \sum_{(x,y)\in I} W(E_t,\eta)E_y \left(\left(x - x_0^{(i)}\right) E_x + \left(y - y_0^{(i)}\right) E_y \right) \end{aligned} \tag{3.9}$$

where h is our time-step. Now this system is a *discrete-time analog system* as opposed to the *continuous-time analog system* we discussed earlier. This implementation method will allow us to put more pixels on the chip at the expense of taking longer to solve the problem.

If we define:

$$A = \sum_{(x,y)\in I} W(E_t,\eta)\mathbf{E_r}\mathbf{E_r}^T \tag{3.10}$$

$$b = \sum_{(x,y)\in I} W(E_t,\eta)\mathbf{E_r}\mathbf{E_r}^T\mathbf{r} \tag{3.11}$$

then the equation that our system will solve is the 2×2 matrix problem:

$$A\mathbf{r}_0 = b \tag{3.12}$$

We can rewrite our solution method into the following form:

$$\mathbf{r}_0^{(i+1)} = \mathbf{r}_0^{(i)} + h\left(b - A\mathbf{r}_0^{(i)}\right) \tag{3.13}$$

This is known as the Richardson iterative method [34] for solving the matrix system in Equation 3.12. It is in fact the simplest iterative method for solving a matrix equation. The transient solution to this equation is:

$$\mathbf{r}_0^{(i)} = \mathbf{e}^{(i)} + \mathbf{r}_0 \tag{3.14}$$

where $\mathbf{r}_0$ is the solution that we want and $\mathbf{e}^{(i)}$ is the error at the ith iteration. This means:

$$\mathbf{r}_0 = A^{-1}b \tag{3.15}$$

$$\mathbf{e}^{(i)} = \mathbf{r}_0^{(i)} - \mathbf{r}_0 = (I - hA)^i \, \mathbf{e}^{(0)} \tag{3.16}$$

Clearly, in order for this system to be stable, we require that the error iterates go to zero. Thus, we must guarantee that the spectral radius of the iteration matrix is less than unity. If we examine the eigenvalues λ' of the iteration matrix we find that they are related to the eigenvalues λ of the matrix A by:

$$\lambda' = 1 - h\lambda \tag{3.17}$$

Since A is symmetric and positive semi-definite (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:

$$0 \leq h \leq \frac{2}{\lambda_{max}} \tag{3.18}$$

We can choose the optimal h to minimize the convergence time of the iteration. This $h_{optimal}$ solves:

$$\min_h \left(\max \left(|1 - h\lambda_{\min}|, |1 - h\lambda_{\max}| \right) \right) \tag{3.19}$$

which gives:

$$h_{optimal} = \frac{2}{\lambda_{\min} + \lambda_{\max}} < \frac{2}{\lambda_{\max}} \tag{3.20}$$

The optimal h depends on the sum $\lambda_{max} + \lambda_{min}$. Since our system is two-dimensional, this is the sum of the eigenvalues which is also the trace of the matrix A. This results in:

$$h_{optimal} = \frac{2}{\sum_{(x,y)\in I} W(E_t, \eta) \left(E_x^2 + E_y^2 \right)} \tag{3.21}$$

Hence, we require our system at a minimum to calculate three quantities: the residue $b - A\mathbf{r}_0^{(i)}$ and the weighted squared image gradient $\sum_{(x,y)\in I} W(E_t, \eta) \left(E_x^2 + E_y^2 \right)$. Additionally, a fourth quantity, $\sum_{(x,y)\in I} |E_t|$, is useful in practice for setting the width η of the weighting function [23].

3.3 The System Architecture

The approach that was decided upon to implement the system described in Equation 3.9 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

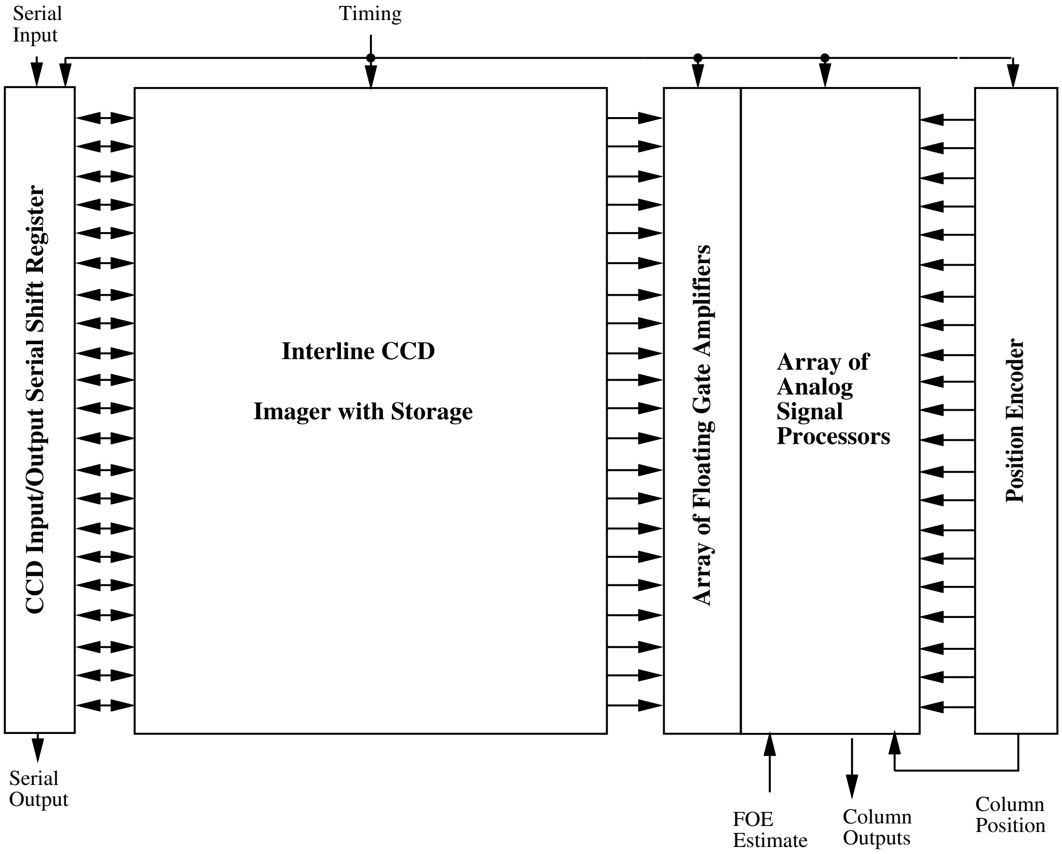


Figure 3-1: Block diagram of the system architecture of the FOE chip.

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.

The system architecture used in the FOE chip is shown in Figure 3-1. 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 possible 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. Of course, the last mode is the most important. 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. Once two images have been acquired, we shift them to the right one column at a time. The floating gate amplifiers transduce this charge signal into voltages which are applied to the analog signal processors. As input, these processors also require the present (x, y) position of the data, provided by the position encoder at the far right of the diagram, and the current estimate of the location of the FOE, $\mathbf{r}_0^{(i)} = (x_0^{(i)}, y_0^{(i)})$, which is driven in from off-chip. From the image data, the pixel position, and the FOE estimate, the processors compute the four desired output currents which are summed up the column in current and sent off-chip. To complete the iterative feedback loop, we must sum these outputs in time as columns of image data are shifted out and update the FOE estimate appropriately. While this could also 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. We could alleviate this problem by moving our imager off-chip and adding a frame buffer, but our architectural goal was a single-chip system.

3.4 Architecture Simulation

Using a constant feedback gain h, Figure 3-2 shows a typical simulated transient for x_0 as it converges to the FOE based on a synthetically generated image sequence. For this simulation, the loop gain $h\lambda_{max}$ is always less than 1 corresponding to an over-damped situation, and the system exhibits the expected exponential decay. Figure 3-3 shows the case where the loop gain is approximately unity, i.e. the critically damped situation, which shows the fastest convergence time. Figure 3-4 demonstrates a under-damped situation where the loop gain is greater than unity but less than 2 where instability sets in, and the transient displays the expected oscillation. Clearly, in spite of the time-varying A matrix and b forcing vector due to new image data on every iteration of the feedback loop, the system qualitatively exhibits the same behavior we expect if A and b were constant. Typical steady-state error was 2% full scale.

In a real system implementation, we will most likely have several significant limitations. The most obvious is that the analog multipliers we use to calculate the residue potentially will be saturating. This means that they will act like multipliers for only a limited input signal range. Outside of this range, they give a fixed signal. Such a limitation was also taken into account in the system simulations. Figure 3-5 shows the same simulation as in Figure 3-3, except the input range of the multipliers is restricted to 15.6% of the available position encoding range, and furthermore the feedback is only allowed to change the estimate by 7.8% full scale per time-step. The initial convergence is linear, as we would expect, until the error signal drops below the saturation threshold, after which it becomes exponential. Based on simulations such as these, we were confident that the FOE system will perform adequately under the expected

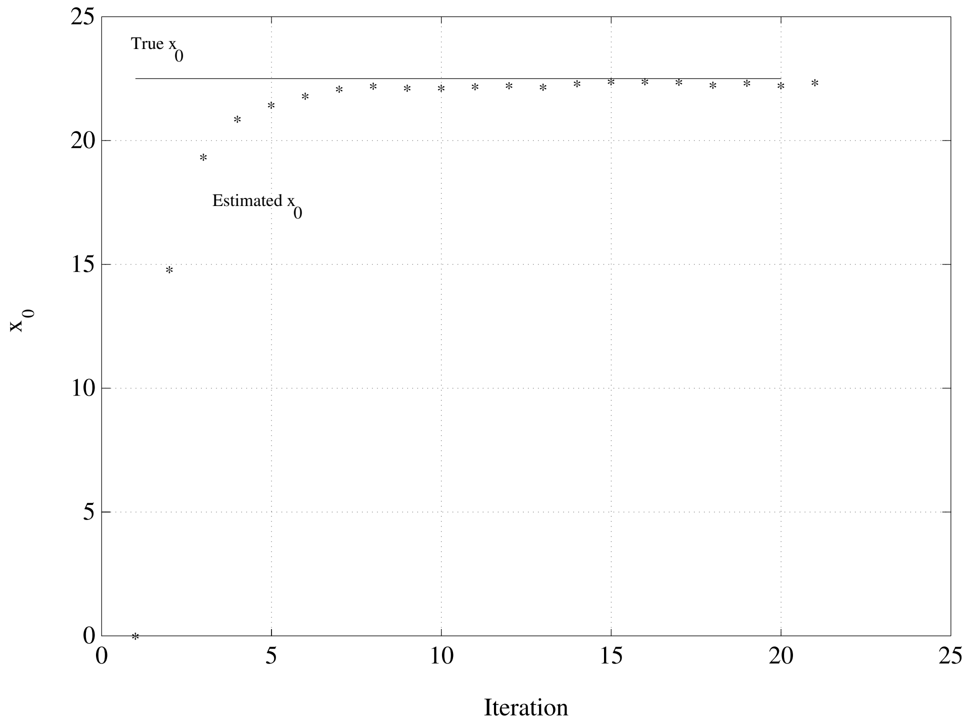


Figure 3-2: Simulated FOE system transient on synthetic image data with $(x_0, y_0) =$ (22.5,42.5). Loop gain $h\lambda_{max}$ is less than 1.

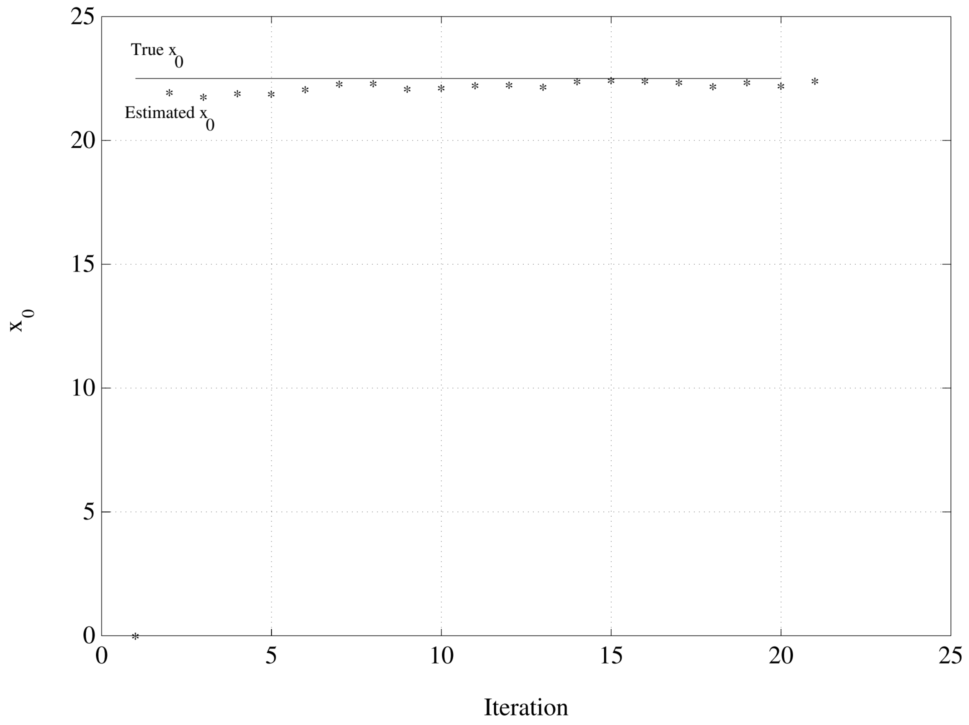


Figure 3-3: Simulated FOE system transient on synthetic image data with $(x_0, y_0) =$ (22.5,42.5). Loop gain $h\lambda_{max} \approx 1$.

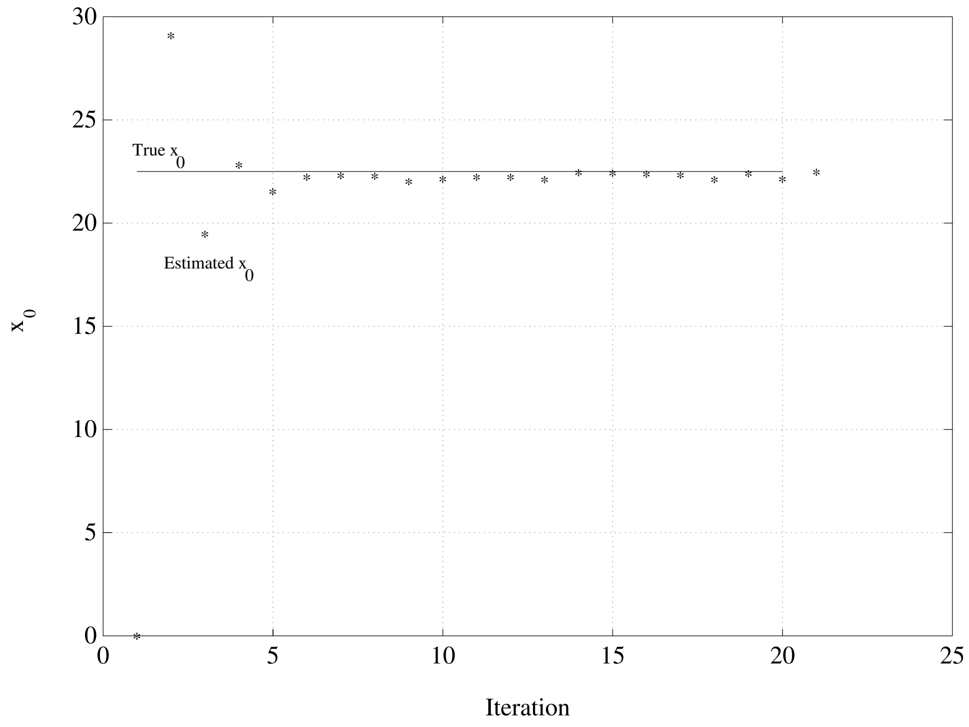


FIGURE 3-4: Simulated FOE system transient on synthetic image data with $(x_0, y_0) =$ (22.5,42.5). Loop gain satisfies: $1 \leq h\lambda_{max} \leq 2$.

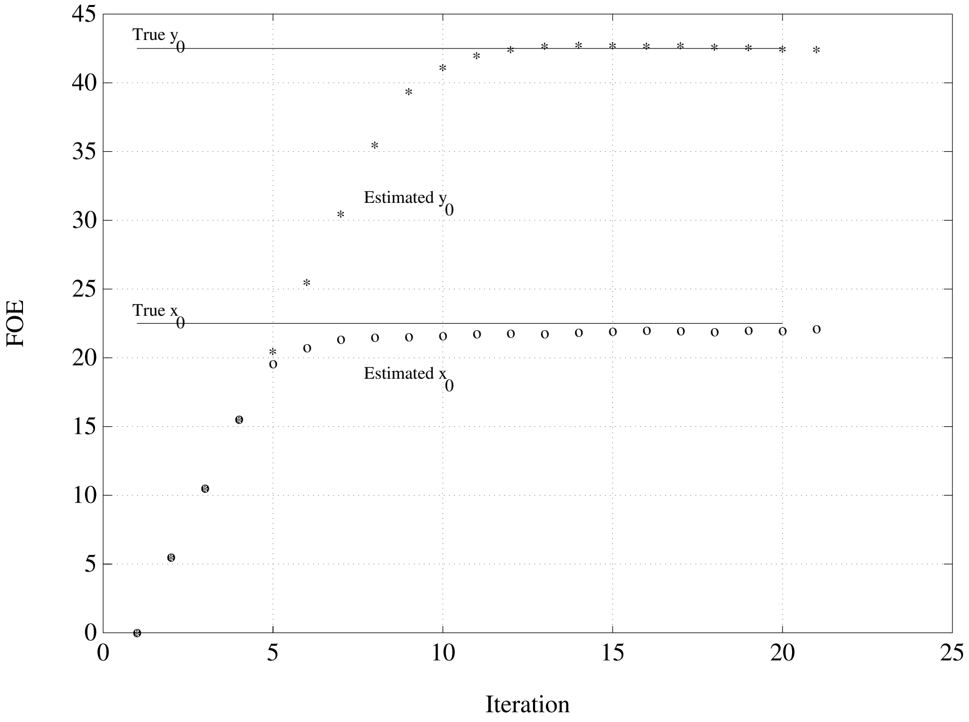


FIGURE 3-5: Simulated FOE system transient on synthetic image data with $(x_0, y_0) =$ (22.5,42.5). Loop gain $h\lambda_{max} \approx 1$. Saturating multipliers and sums are modeled.

conditions of varying A,b and saturating multipliers.

3.5 Circuit Structures

With the specified system architecture in mind, we can now turn to an explicit description of the various circuit components in the FOE chip. These can be broken up into two parts, each of which will be discussed in turn: the CCD section comprising the imager, I/O register, and floating gate amplifiers, and the CMOS section comprising the processing array and the position encoder. Figure 3-6 shows a photograph of the fabricated FOE chip with the various sections highlighted. The floorplan of the actual chip closely matches the system architecture shown in Figure 3-1.

3.5.1 Charge Coupled Device Fundamentals

In this section, we present a short review of the principles of charge coupled devices which then segues into the design of the interline CCD imager used for image acquisition and the floating gate amplifiers used for charge sensing on the FOE chip. There are two basic types of CCD structures: the surface channel CCD where the charge signal resides at the surface of the semiconductor and oxide interface and the buried channel CCD where the charge is confined to a thin layer away from the interface [35]. The process which we will be using to fabricate the FOE chip is an 10V n-well $2\mu m$ buried channel CCD/BiCMOS double-poly double-metal Orbit process, which we access through MOSIS [36]. However, we begin our discussion by focusing on the surface channel device due to its conceptual simplicity.

3.5.1.1 The MOS Capacitor

The surface channel CCD bears a direct correspondence with the MOS capacitor. The basic MOS capacitor structure, shown in Figure 3-7, is a sandwich consisting of a gate electrode over a thin insulator on top of a silicon semiconductor substrate. Typically, the gate electrode is composed of degenerately doped polysilicon and the insulator of silicon dioxide. In this discussion, we assume that the silicon semiconductor substrate is of p-type. The various equilibrium states of the MOS capacitor can be summarized in the energy band and charge distribution diagrams of Figure 3-8. If a negative voltage is applied to the gate relative to the bulk (a), then holes (the majority carrier in p-type material) are attracted to the surface of the semiconductor, imaging the negative gate charge. Due to band bending, the semiconductor surface is even more p-type than the bulk, and is said to be *accumulated* with majority carriers. When no bias is applied (b), which is the *flat-band* condition, then there will be a uniform distribution of holes throughout the semiconductor, and none attracted to the interface. Because of oxide charge and differences in the work functions between the gate and the substrate, a real device requires a nonzero gate voltage V_{FB} for this flat-band condition. As the gate voltage is raised

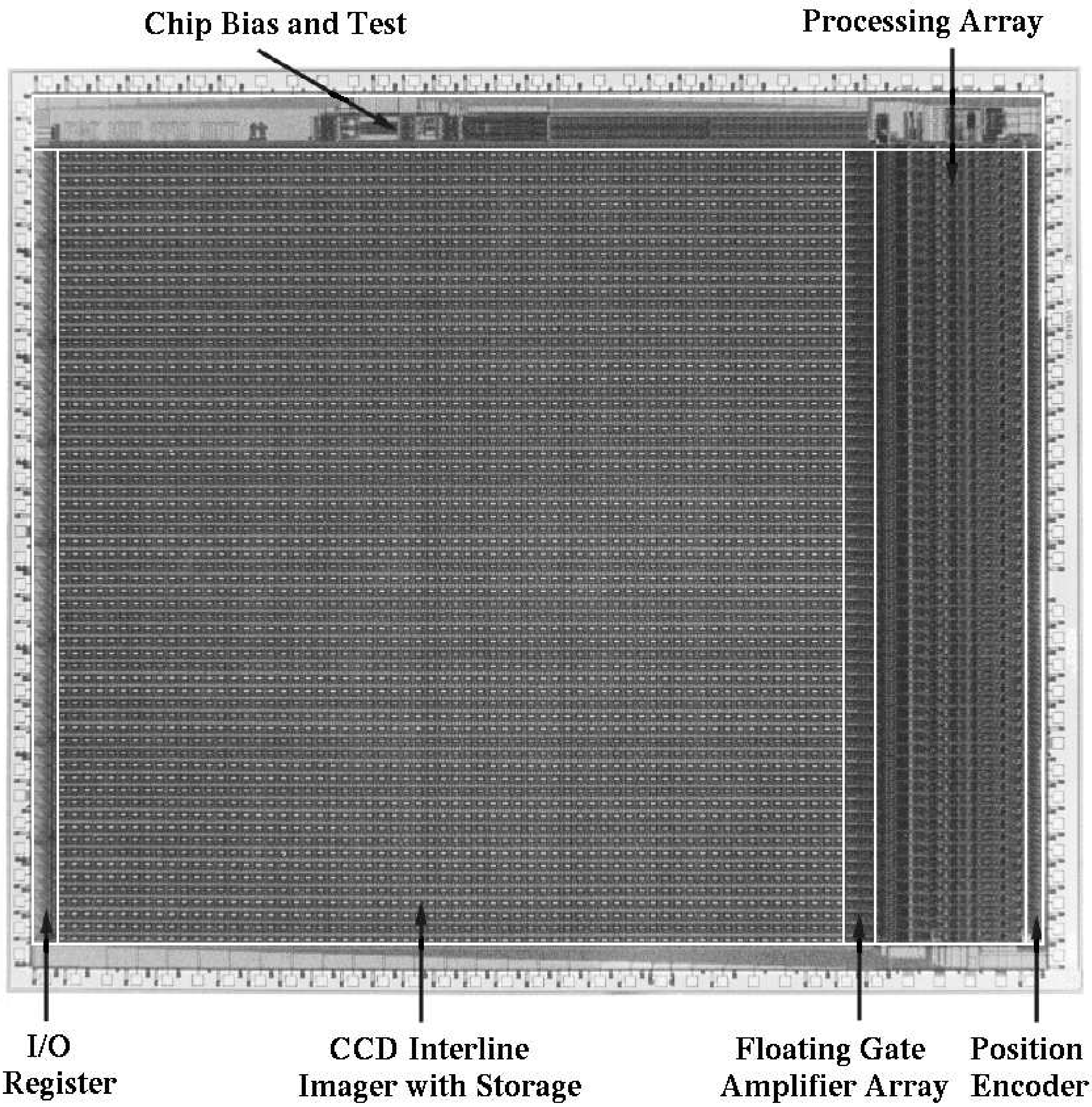


FIGURE 3-6: Photograph of the fabricated FOE chip with the various components indicated.

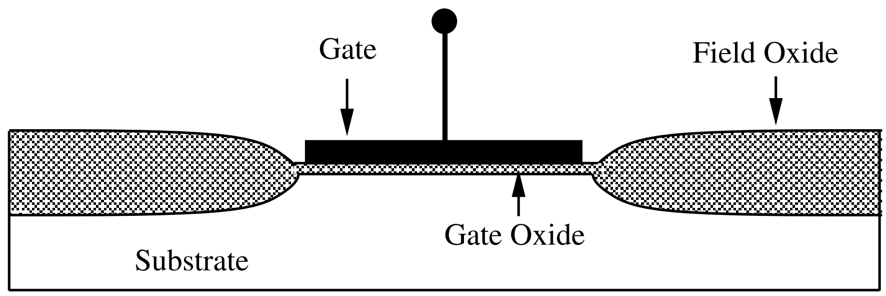


FIGURE 3-7: An MOS Capacitor.

above flat-band (c), holes are repelled from the surface and a depletion layer is formed exposing negative ionized acceptor charge. As the gate voltage continues to be increased, this depletion layer extends further and further into the bulk and the surface potential of the semiconductor simultaneously becomes more and more positive. A gate voltage is eventually reached wherein electrons (the minority carriers in p-type material) are attracted to the surface, forming a thin layer of *inversion charge*. This occurs when the surface potential of the semiconductor ϕ_s is between ϕ_F and $2\phi_F$, where $\phi_F = (kT/q)\ln(N_a/n_i)$. In this *weak inversion* regime (d), further increases in the gate bias attracts more inversion charge and also increases the width of the depletion region. The conduction band bends close enough to the Fermi level such that appreciable amounts of minority electrons are produced. Finally, a gate bias is reached after which further increases result in more inversion charge, while the depletion width remains essentially unchanged. This is the *strong-inversion* regime ($\phi_s \approx 2\phi_F$). The onset of this regime (e) is typically defined when the Fermi level is as much above the intrinsic level at the surface as it is below the intrinsic level in the bulk.

The formation of an inversion layer in an MOS transistor normally corresponds to the creation of a conducting channel from source to drain with the minority charge coming from the source and drain of the transistor. However, in the MOS capacitor there are no drain and source providing minority carriers; the inversion charge typically comes from either thermal generation, optical processes, or is intentionally injected into the device by a charge input structure, which we will discuss later. Hence, if the gate of the MOS capacitor is suddenly stepped well beyond the threshold voltage, an inversion layer cannot form immediately. In this situation, called *deep-depletion*, the depletion region extends much further into the bulk than it would normally at equilibrium, and most of the gate-to-bulk voltage is dropped across it. As minority carriers are introduced, the surface potential drops and the inversion layer increases until equilibrium is finally attained. However, the MOS capacitor is seldom allowed to achieve equilibrium when operated as a CCD ; it is typically operated in the non-equilibrium deep-depletion mode. It is normally used as a charge storage device by intentionally introducing inversion charge to the deep-depleted substrate beneath the gate.

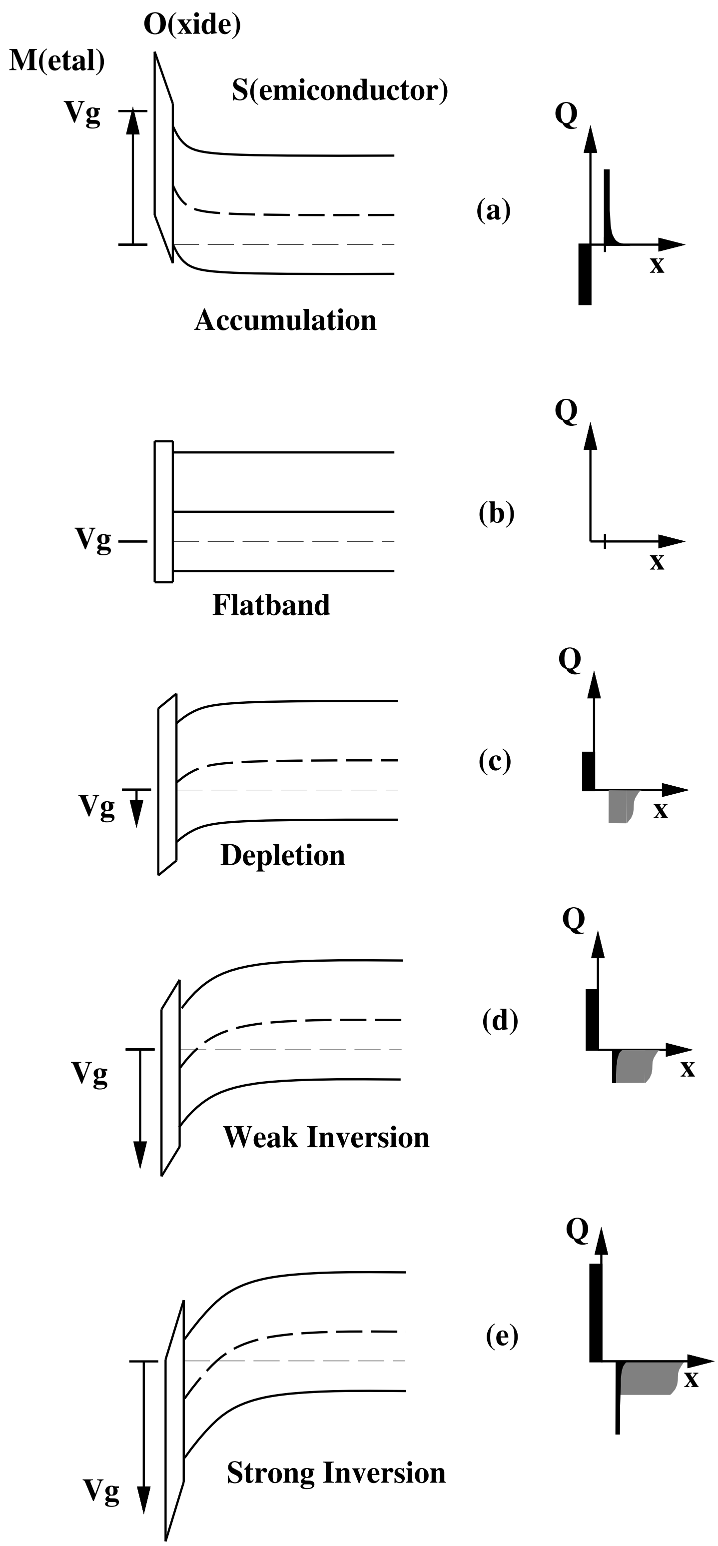


FIGURE 3-8: Energy band diagrams and charge distributions for the MOS capacitor in (a) accumulation, (b) flatband, (c) depletion, (d) weak inversion, and (e) strong inversion. After [37].

3.5.1.2 The Delta-Depletion Approximation

The *delta-depletion* approximation [37] is often used to derive a useful closed form solution relating the surface potential with the gate to bulk voltage and inversion charge. In this approximation, the inversion charge is assumed to be a delta function at the semiconductor surface, while the space-charge layer is assumed to be fully depleted of mobile carriers and ending abruptly at a depth $x = W$. In the completely depleted case, we have the charge density ρ:

$$\rho = -q(N_a - N_d) \text{ for } 0 \leq x \leq W \tag{3.22}$$

Integrating once to get the $E-$field with the boundary condition of zero $E-$field in the bulk, we find:

$$E = \left(\frac{q(N_a - N_d)}{\epsilon_{si}}\right)(W - x) \text{ for } 0 \leq x \leq W \tag{3.23}$$

Integrating once again to get potential with the boundary condition of zero potential in the bulk:

$$\phi = \left(\frac{q(N_a - N_d)}{2\epsilon_{si}}\right)(W - x)^2 \text{ for } 0 \leq x \leq W \tag{3.24}$$

The surface potential is $\phi(0)$:

$$\phi_s = \left(\frac{q(N_a - N_d)}{2\epsilon_{si}}\right)W^2 \tag{3.25}$$

Turning this around we find that the depletion width is

$$W = \sqrt{\frac{2\epsilon_{si}\phi_s}{q(N_a - N_d)}} \tag{3.26}$$

The bias voltage $V_G - V_{FB}$ is dropped across the oxide (V_{ox}) and the substrate (ϕ_s) while the total semiconductor charge Q_s is the sum of the inversion charge Q_{inv} and the bulk charge Q_B:

$$\begin{aligned}
V_G - V_{FB} &= V_{ox} + \phi_s && (3.27)\\
&= -\left(\frac{Q_s}{C_{ox}}\right) + \phi_s && (3.28)\\
&= -\left(\frac{Q_{inv} + Q_B}{C_{ox}}\right) + \phi_s && (3.29)\\
&= -\left(\frac{Q_{inv} + q(N_a - N_d)W}{C_{ox}}\right) + \phi_s && (3.30)
\end{aligned}$$

Substituting in for the depletion width from Equation 3.26 and solving, we find:

$$\phi_s = V_G - V_{FB} + \frac{Q_{inv}}{C_{ox}} - V_o\left[\sqrt{1 + 2\left(\frac{V_G - V_{FB} + Q_{inv}/C_{ox}}{V_o}\right)} - 1\right] \tag{3.31}$$

where

$$V_o = \frac{q(N_a - N_d)\epsilon_{si}}{C_{ox}^2} \tag{3.32}$$

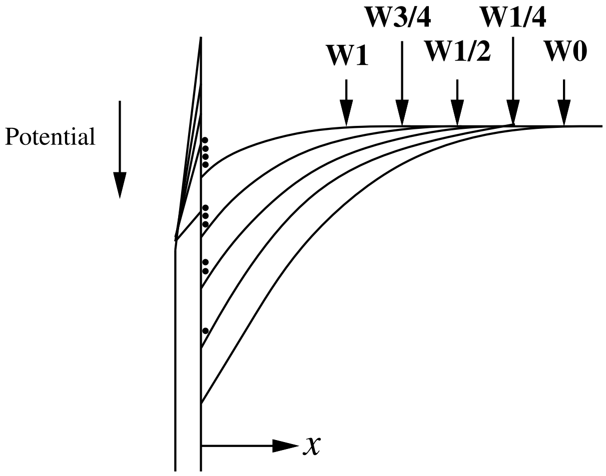


FIGURE 3-9: The potential into the semiconductor for a deep-depletion MOS capacitor with 0, 1/4, 1/2, 3/4, and full charge. The Ws refer to the widths of the resulting depletion regions. After [37].

Typically, in today's processes V_o is small and neglecting the last term usually leads to $< 10\%$ error. As illustrated in Figure 3-9, this leaves a linear relationship between the surface potential ϕ_s, the inversion charge Q_{inv}, and the gate voltage V_g:

$$\phi_s + V_{FB} \approx V_G + \frac{Q_{inv}}{C_{ox}} \tag{3.33}$$

3.5.1.3 The Potential Well Analogy

The linear nature of this relationship allows us to form the highly useful *potential well* analogy with regard to the charge-storage operation of the MOS capacitor. Applying a gate voltage in excess of the threshold voltage causes the formation of a "potential well" linearly related to the gate voltage V_G. As shown in Figure 3-10, the surface potential can then be viewed as describing the location of the surface of the "water" (charge) in this well. Introduction of inversion charge into the well or *bucket* causes it to fill up; the depth of the well as measured from the top to the surface of the water (charge) decreases analogous to the decrease in surface potential with increasing inversion charge in Equation 3.33 ($Q_{inv} < 0$ for p-type material). Of course, this simple analogy of water filling up buckets created by the gate voltage is for conceptional purpose only; the inversion charge is really contained in a thin layer at the oxide-semiconductor interface. Also, the bucket cannot be completely filled up; it has a minimum depth of $2\phi_F$. The maximum charge handling capacity of the MOS capacitor is fixed by the depth of the well, determined by both the thickness of the gate oxide and the magnitude of the gate voltage, and the area of the gate, which determines the area of the cross-section of the capacitor. Lateral charge confinement is typically accomplished by potential barriers provided

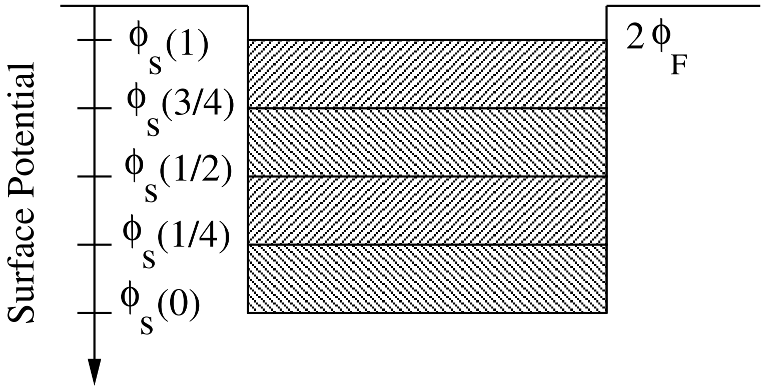


Figure 3-10: The potential well analogy showing the surface potential and the corresponding "bucket" filled with "water" at the 0, 1/4, 1/2, 3/4, and full packet levels. After [37].

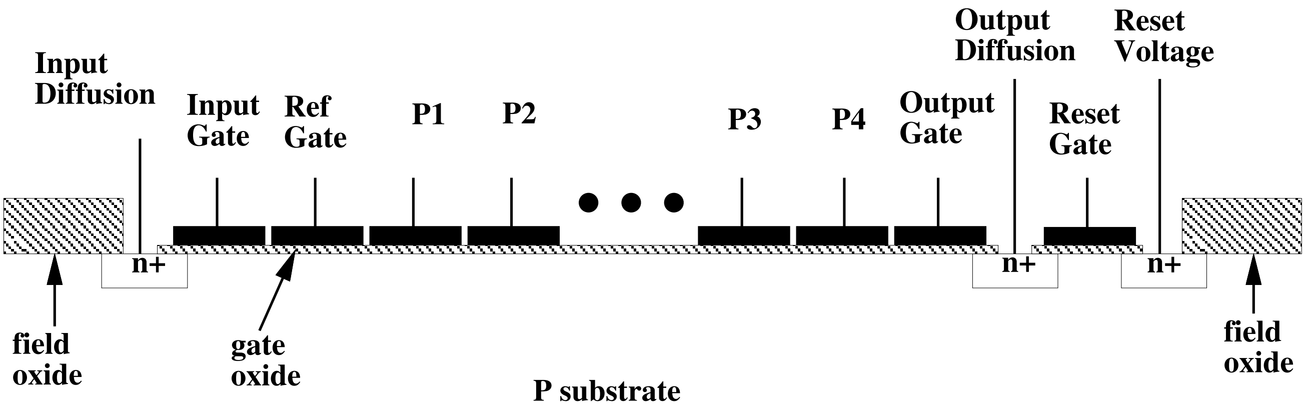


Figure 3-11: Diagrammatic representation of a typical surface-channel ccd, including a fill-and-spill input structure on the left and a floating diffusion output structure on the right.

by field implants and/or adjacent gates. To find Q_{max}, we substitute the strong inversion condition $\phi_s = 2\phi_F$ and rearrange Equation 3.33 to find:

$$Q_{max} \approx -(V_G - V_{FB} - 2\phi_F)C_{ox} \tag{3.34}$$

3.5.1.4 The Surface Channel CCD

A surface channel charge-coupled device (ccd) is an array of closely spaced mos capacitors as shown in Figure 3-11 [35, 38, 39, 37]. The signal of interest consists of "charge" stored in the depleted silicon beneath the gates. Each mos capacitor is "coupled" to its immediate neighbors through the substrate, which forms a common bottom plate for the capacitors. Such a structure has traditionally had three main applications: as a method for short-term storage (for example, dynamic memory), as a shift register, and as an image sensor. Other applications such as charge domain signal processing have received much attention [38, 39, 40, 41, 42, 6, 12, 8, 14] but have yet to achieve significant commercial success.

3.5.1.5 Charge Storage and Dark Current

The charge storage capability of the MOS capacitor obviously leads to the first application. The storage is only short-term as charge cannot be held indefinitely under the gate. The potential wells formed beneath the CCD gates will eventually fill up on their own with thermally generated carriers, called dark current. However, with today's processing technology the thermal relaxation time of CCDs can be on the order of minutes at room temperature [39]. The figure of merit for dark current is the thermal current density J_D, and by optimizing fabrication to minimize dark current, current densities of $J_D = 1 - 2nA/cm^2$ are achieved by modern CCD processes. With typical numbers such as a t_{ox} of $40nm$, $C_{ox} = 0.86fF/\mu m^2$, and a $5V$ bias, $Q_{max} \approx 2.5 \times 10^{12} e^-/cm^2$ (which is about $2.5Me-$ for a CCD of size $100\mu m^2$). With a J_D of $1nA/cm^2$, it would take on order 400 seconds to form a full inversion charge. Of course, for eight bit accuracy this would be reduced to 3 seconds for 1 LSB of offset. For CCD systems operating in the multiple MHz range, the error due to dark current is negligible. Of course, dark current is a thermal process, and as such exhibits exponential dependence on temperature. As a rule of thumb, one expects a doubling of J_D for every $6°C$ temperature increase.

3.5.1.6 CCDs as Analog Shift Registers

By suitable manipulation of gate voltages, we can transfer charge from beneath a gate to an adjacent gate. This is the fundamental operation of an analog shift register, the second main CCD application. Figure 3-12 provides an example of this shifting of charge in CCDs using the bucket and water analogy. Here, we use four phases of clock for moving the charge down the register. Other clocking schemes are certainly possible. Two phase and three phase clocking are popular alternatives. However, two phase clocking uses a buried electrode or castellated gate oxide to accomplish charge transfer, requiring additional processing [38]. Three phase clocking normally requires three levels of polysilicon. Since we will be using a double polysilicon process for the FOE chip, we cannot use standard three phase clocking, as that would result in each clock being connected to *both* levels of polysilicon. In this case we could not use clock levels to compensate for mismatches in the CCD operation due to the different thicknesses of gate oxide under each polysilicon. Four-phase clocking allows for this sort of compensation, and this methodology stores charge under two gates at a time, effectively doubling the size of the maximum charge packet when compared to the two-phase or three-phase approach.

Initially (time t_1), the charge is confined under ϕ_1 and ϕ_2, while the other two phase are held low. At time t_2, we raise the gate voltage on ϕ_3 and drop it on ϕ_1. This causes the potential well under ϕ_1 to collapse and a potential well under ϕ_3 to form. The charge flows into the newly created well and out of the collapsed well, so that at time t_3 the charge now resides under phases ϕ_2 and ϕ_3. This process of transferring charge from under a gate to an adjacent

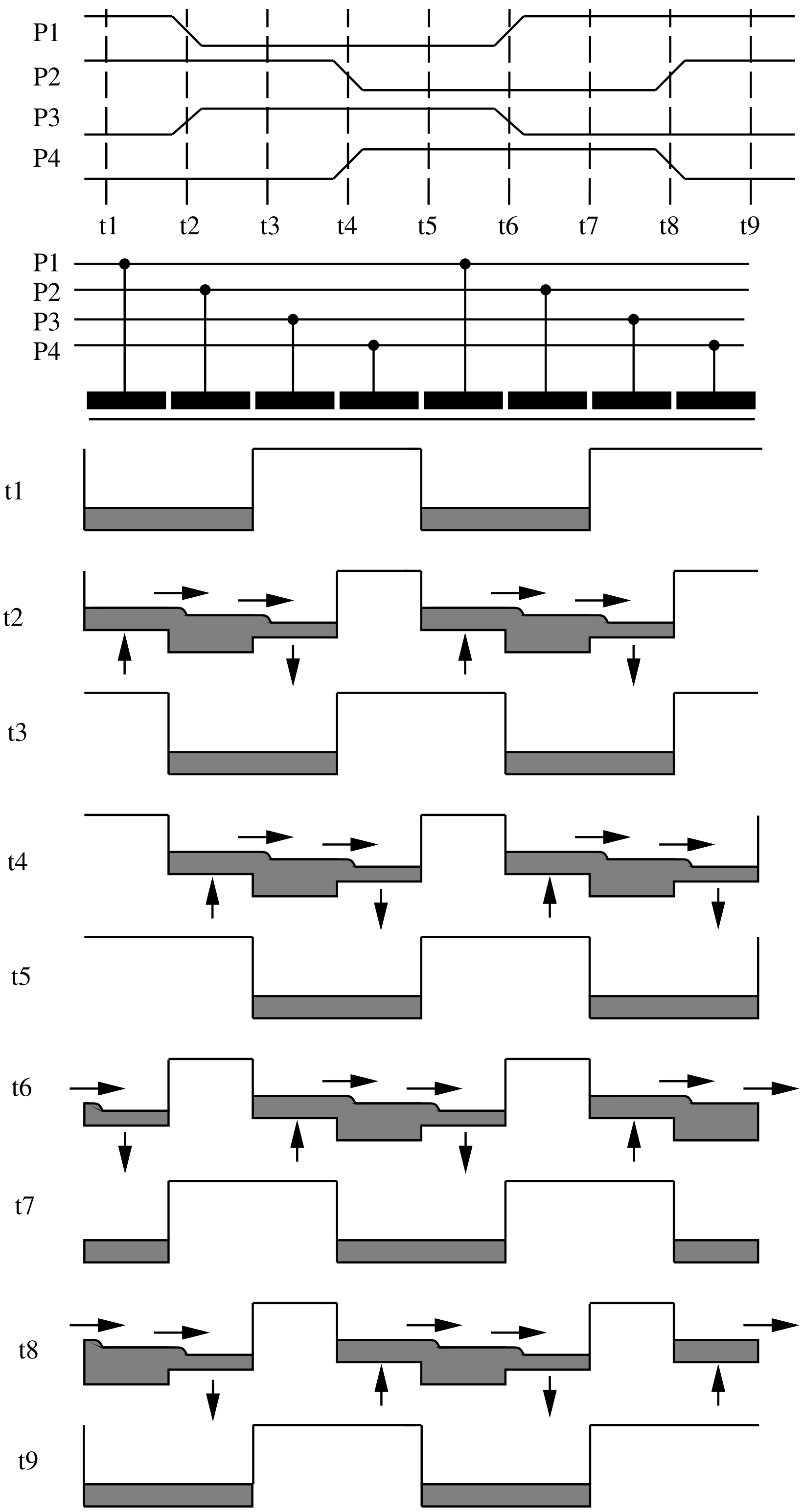


FIGURE 3-12: Shifting charge using two levels of polysilicon and four overlapping clock phases.

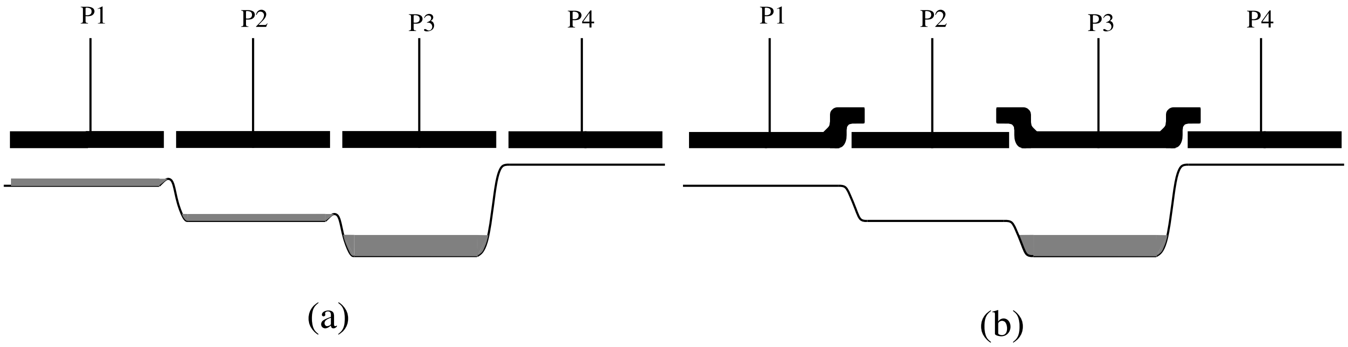


FIGURE 3-13: (a) Potential "bumps" caused by gate separation leading to transfer inefficiency. (b) Overlapping gate structure alleviating potential "bumps".

gate is accomplished through three mechanisms. Initially, the charge moves due to self-induced drift caused by the electrostatic repulsion of the carriers. In the intermediate stages of transfer, the lateral fringing fields in the substrate set up by the potential difference between the adjacent gates dominates the transfer. The final stage is governed by thermal diffusion of the carriers.

Proceeding in a likewise *bucket-brigade* manner, we can shift charge from ϕ_2, ϕ_3 to ϕ_3, ϕ_4 at time t_5 to ϕ_4, ϕ_1 at time t_7 and back to ϕ_1, ϕ_2 at time t_9. Note that this process can be reversed and the charge made to flow to the left instead of the right. Thus, the shifting can be bidirectional.

3.5.1.7 Transfer Inefficiency

Of critical importance for the proper operation of CCDs is how completely charge can be transferred from under a gate to its adjacent neighbor. The amount of charge lost in each transfer is termed the transfer inefficiency. There are three main causes for the loss of transfer efficiency: a) potential barriers between the gates, b) insufficient time was allowed for the transfer process to complete, and c) some charge is trapped in interface states whose emission time is longer than the transfer time. If the gates are spaced too far apart, then the surface potential develops bumps as shown in Figure 3-13a, trapping charge and seriously degrading performance. To alleviate this, adjacent CCDs have overlapping gates as shown in Figure 3-13b in order to assure proper transfer.

Charge-transport theory, neglecting interface trapping, predicts that the characteristic time of charge transfer goes as L^2, where L is the length of the CCD gate [37]. This would seem to imply that any desired transfer efficiency can be achieved if one only waited long enough. This turns out to not be the case due to surface traps, which become the limiting factor for charge transfer in surface channel CCDs. One technique for reducing the effect of these traps is to use *fat zeros*. In this approach, the CCD is operated with a small amount of charge always contained in every well, and the signal charge is then added to this background level. The idea is that the traps are kept full by the fat zero, reducing their effect on the charge packet size.

Even so, good surface channel CCDs can achieve transfer inefficiencies on order 1×10^{-4}, which gives a transfer efficiency of 0.9999. The reason that having a very high transfer efficiency is so important is due to the large number of transfers that a charge packet is typically subjected to, which may number in the thousands. A typical large scale CCD imager may require a thousand or more transfers worst case before the last charge packet in the array is sensed, and with four nines of transfer efficiency, one thousand transfers results in $(0.9999)^{1000} = 0.90$ giving a 10% signal reduction.

3.5.1.8 Imaging

CCDs have by far had their most impact as image sensors [43, 38, 37]. Upon entering the substrate, photons are absorbed and create electron-hole pairs. This absorption process is most effective in the visible and near-infrared wavelengths for silicon. The minority carriers (electrons in p-type material) generated in the depletion region are all collected there, while those within a diffusion length of the depletion region experience a diffusion gradient towards the region, enhancing their collection. Electrons generated outside of one diffusion length are highly likely to be recombined and lost. Once in the depletion region of the CCD structure, the electrons are attracted by those gates with the biggest surface potentials, and collect there. The efficiency of this operation, the quantum efficiency, is defined as the mean number of minority carriers produced by a photon and collected by the CCD.

There are two main methods for introduction of photons into the substrate: through the front-side or the back-side of the imaging chip. Illuminating through the front has as its advantage simplicity, while back-side illumination frequently requires thinning of the chip. The drawback of front-side illumination is the resulting interference effects in the visible spectrum due to the sandwich of oxide and polysilicon layers which the photons must pass through to reach the substrate. This produces non-uniform responsivity with wavelength, and also reduces the quantum efficiency substantially as shown in Figure 3-14. The process which we will use to fabricate the FOE chip does not have the required thinning, nor does the package provided by MOSIS allow for backside illumination. Hence, we will use front-face imaging, with light passing through openings in a second-level metal light shield on the chip and then through the polysilicon gate of a collection CCD to the substrate. This typically has a quantum efficiency of approximately 20-30%.

3.5.1.9 Charge Input

Signal charge under a CCD gate can arise from imaging or thermal generation. Obviously, we would like to be able to intentionally introduce a known amount of charge into a CCD structure. The charge-input circuit shown in Figure 3-15a uses the traditional *fill-and-spill* technique [38]. This structure consists of an input diffusion ID forming a diode with the substrate and two input gates, IS and RG, followed by a standard CCD shift register. Initially (time t_1), the diode

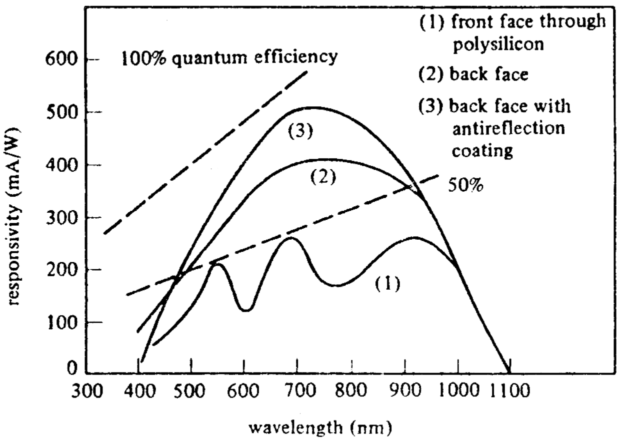


FIGURE 3-14: Typical optical CCD responsivity. From [38].

is reverse biased, and the two gates have a potential difference representing the desired input. At time t_2 the diode voltage is pulsed low filling the potential well created by the two input gates. At time t_3, the diode is sent high, once again becoming reverse biased and allowing the excess charge to spill back. This fill-and-spill method leaves a charge packet under the input gates which is proportional to their potential difference, and which can then be entered into the CCD shift register at time t_4.

There are two main drawbacks to the standard fill-and-spill structure. The voltage on RG is typically held constant, and in order to get the charge out of RG and into the shift register at time t_4 the potential well in the register must be deeper. The result of this is a reduction in the size of the charge packet that can be input into the structure. To fix this, the reference gate can be pulsed low to get the charge out. The other main drawback is that the gates doing the metering are adjacent, and therefore must be on different levels of polysilicon because of the required overlap for transfer efficiency. Typically, the two polys have slightly different characteristics such as gate oxide thickness t_{ox}, oxide charge, etc. This gives rise to a difference in the depth of the potential well for the same voltage. To fix these problems, we can use the so-called split-gate input structure [44]. This circuit is a slight modification of the fill-and-spill structure and is shown in Figure 3-15b.

Here we have added a floating diffusion in between the metering gates. At time t_1, the diode is initially reverse biased, the input gate IS is fixed at the desired input level, the reference gate RG is off, and the first gate of the shift register P_1 is on with an empty well. At time t_2 the diode voltage is pulsed low filling the floating diffusion up to the level defined by the input gate potential. At time t_3, the diode is sent high, once again becoming reverse biased and allowing the excess charge to spill back. Signal charge is now stored in the floating diffusion. At time t_4, we pulse RG to the reference level, and a charge packet proportional to the voltage difference between the input gate and the reference level is entered into the shift register. Now, the two

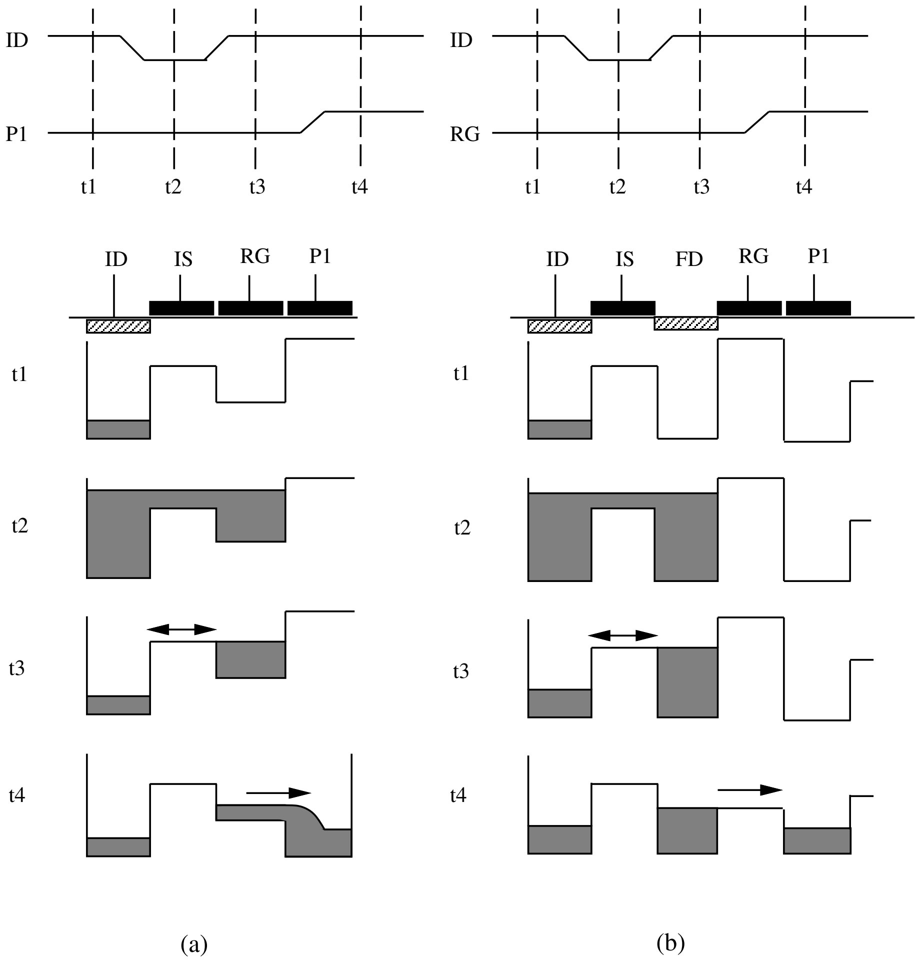


Figure 3-15: Input structures, clocking waveforms, surface potentials, and signal charge, for (a) the standard fill-and-spill approach and (b) for the split-gate fill-and-spill technique.

metering gates are separated by the floating diffusion and can be fabricated in the same poly.

3.5.1.10 Charge Output

Inputting charge is only half of the game. We also need some way of sensing the charge in a CCD. The simplest output structure for doing so is shown in Figure 3-16a, while the clock waveforms for driving the structure and the output are shown in Figure 3-17a. The floating diffusion output structure consists of an output gate followed by an output diffusion. The voltage of this diffusion is buffered by a source-follower, forming the output. In this method of charge sensing, we pre-charge the diffusion to the reset voltage VR, leave it floating, and then clock the charge into the diffusion, resulting in the voltage change on the waveform as shown in Figure 3-17a. At t_1, we assume that the charge is under the gates of clock phases ϕ_3 and ϕ_4 while the output gate is blocking the output diffusion, which is being pre-charged by the reset transistor. At t_2, we transfer the charge entirely under ϕ_4. At prior stages in the shift register,

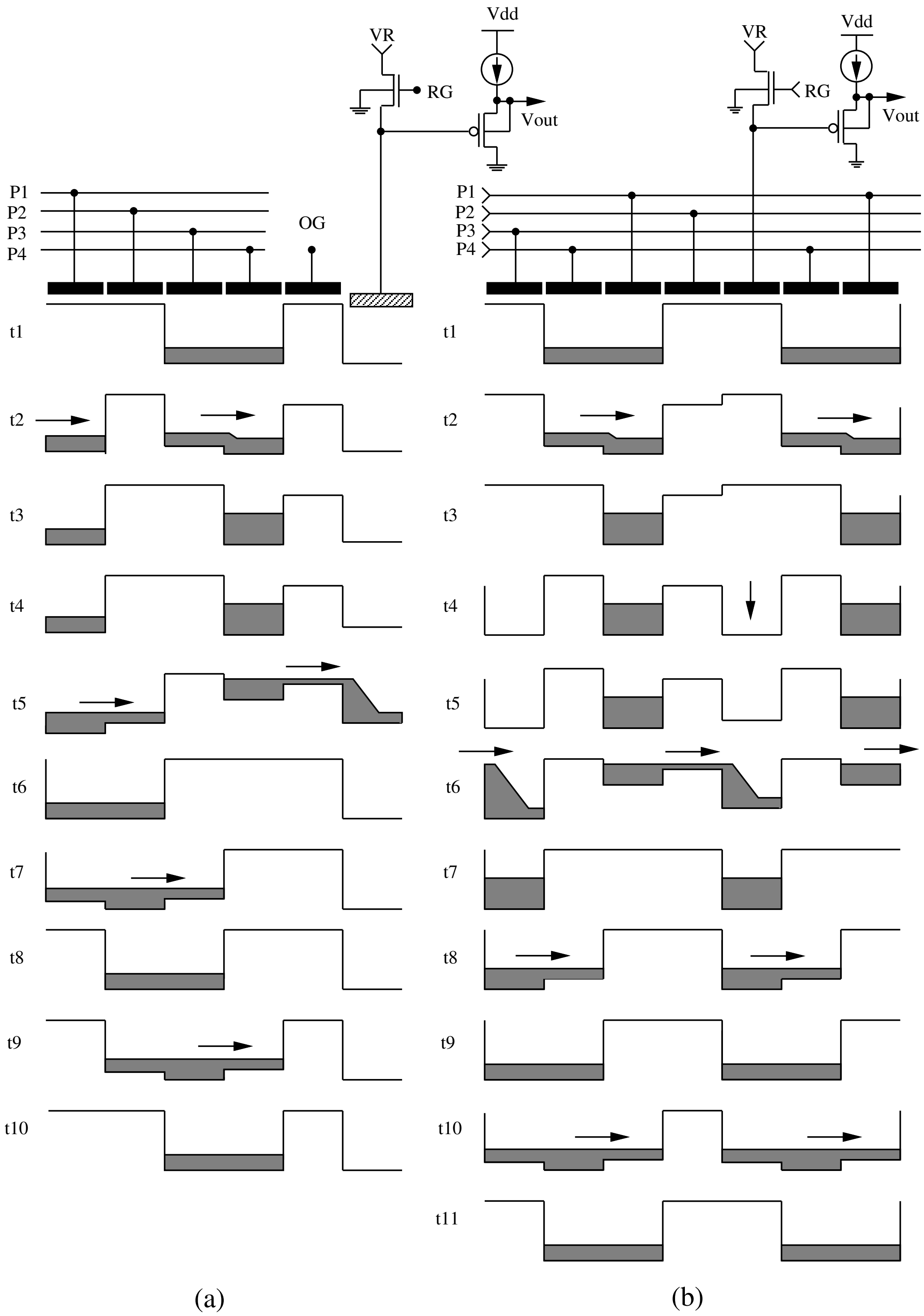


FIGURE 3-16: Potentials and charge locations during shifting and sensing for (a) the floating diffusion output structure and (b) the floating gate output structure.

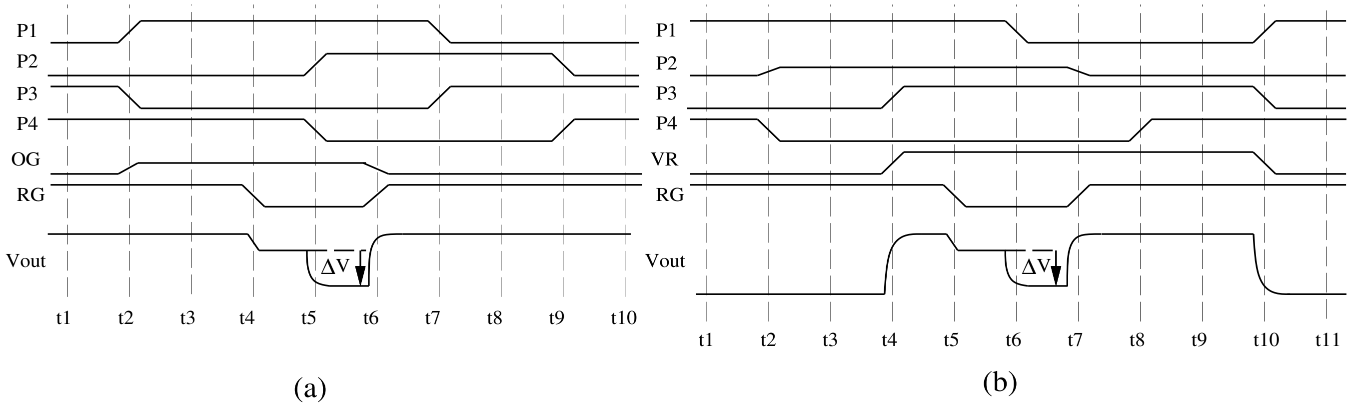


FIGURE 3-17: Required clock waveforms for output using (a) the floating diffusion output structure and (b) the floating gate output structure.

we also send phase ϕ_1 high, accomplishing a ϕ_3, ϕ_4 to ϕ_4, ϕ_1 transfer by t_3. At t_2 we have also set the voltage on the output gate slightly higher than its off state, as we are going to transfer the charge from ϕ_4 *through the output gate* and into the diffusion. The reason we use this *DC blocking gate* technique is because there is significant capacitive coupling between the output gate and diffusion. Hence, we cannot let the output gate voltage change during the charge sense. At t_4 we turn off the reset transistor leaving the output diffusion floating, and at t_5 we transfer the charge through the blocking gate and into the diffusion. In prior stages of the shift register, we also perform the ϕ_4, ϕ_1 to ϕ_1, ϕ_2 transfer at the same time. The source follower is allowed to settle and the output used, after which the diffusion is once again reset. Finally, during t_6 through t_{10}, we shift charge from ϕ_1, ϕ_2 to ϕ_2, ϕ_3 and then to ϕ_3, ϕ_4 completing the output cycle.

There is one major drawback in using a floating diffusion to sense the output charge. This technique of charge sensing is inherently destructive in that we have extracted the charge and cannot sense it again. In the FOE chip, we need to sense the same charge packet several times over in order to use first centered differencing to estimate image brightness gradients. Figure 3-16b shows an output structure for the non-destructive sensing of charge, while Figure 3-17b shows the driving waveforms and the resultant output. Here, we are using a gate for the sense as opposed to the diffusion. The idea here is to use a similar technique as in the floating diffusion case, except now we rely on the capacitive coupling to a gate for the charge sense [45, 46]. The advantage of this approach is that the charge is not removed in the process and can continue on down the shift register, where there may be more floating gate amplifiers. At t_1, we assume that the charge is under the gates of clock phases ϕ_4 and ϕ_1. At t_2, we transfer the charge entirely under ϕ_1 and set up ϕ_2 as a DC blocking gate. At t_4 we pre-charge the floating gate high and turn off the reset transistor at t_5 leaving the gate floating. At t_6 we transfer the charge from ϕ_1 through ϕ_2 acting as a blocking gate and into the floating gate. In prior stages of the shift register this has the effect of transferring the charge into ϕ_3. The

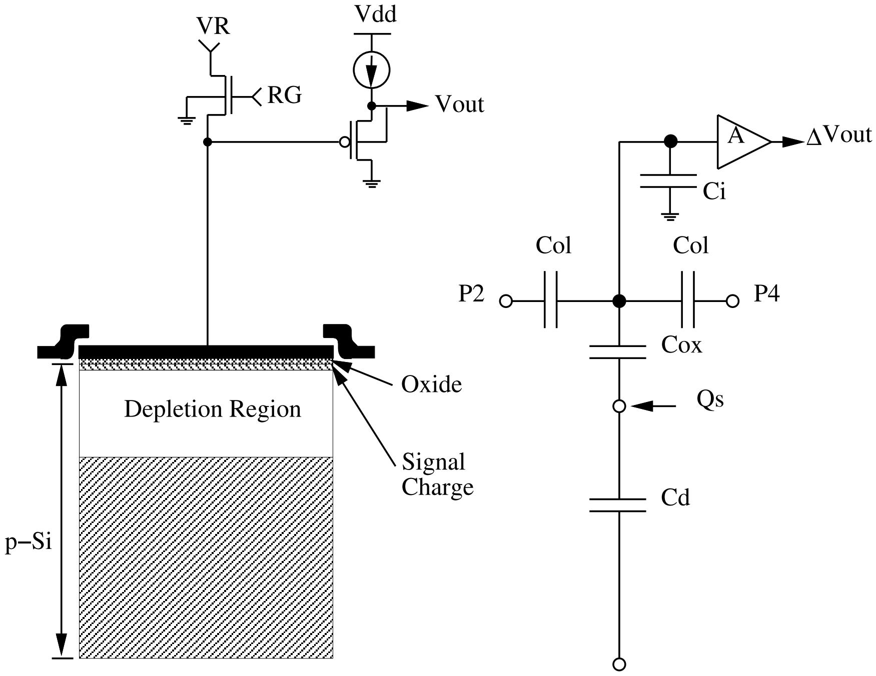


FIGURE 3-18: Lumped circuit model for the floating gate output structure in a surface channel CCD.

source follower is allowed to settle and the output used, after which the gate is once again reset at t_7. Finally, during t_8 through t_{11}, we shift charge from ϕ_3 to ϕ_3, ϕ_4 and from there to ϕ_4, ϕ_1, completing the output cycle.

Figure 3-18 gives a lumped circuit element model for the floating gate sense output where we have explicitly shown all the significant capacitances involved. Charge is injected at the semiconductor oxide interface. There is a depletion capacitance C_D to the substrate as well as the oxide capacitance C_{ox} to the floating gate. Coupling to the floating gate are the overlap capacitances C_{ol} from the adjacent gates, as well as the capacitance C_i representing the input capacitance of the source follower and the reset transistor. We note that:

$$\Delta V_{out} = \Delta V_{ox} + \Delta V_d \tag{3.35}$$

Writing this in terms of charge gives:

$$\frac{\Delta Q_l}{C_l} = \frac{\Delta Q_{ox}}{C_{ox}} + \frac{\Delta Q_d}{C_d} \tag{3.36}$$

where we have lumped together $C_l = 2C_{ol} + C_i$. By charge balance at the output node, we have $\Delta Q_l + \Delta Q_{ox} = 0$, and therefore:

$$\Delta Q_l \left(\frac{1}{C_l} + \frac{1}{C_{ox}} \right) = \frac{\Delta Q_d}{C_d} \tag{3.37}$$

By charge balance at the injection point, we have $Q_s + \Delta Q_{ox} - \Delta Q_d = 0$ and hence:

$$Q_s = \Delta Q_d - \Delta Q_{ox} \tag{3.38}$$

$$= \Delta Q_d + \Delta Q_l \tag{3.39}$$

$$= \left[1 + C_d\left(\frac{1}{C_l} + \frac{1}{C_{ox}}\right)\right]\Delta Q_l \tag{3.40}$$

Since $\Delta Q_l = C_l \Delta V_{out}$, we have the result:

$$\Delta V_{out} = \frac{Q_s}{C_l}\left[\frac{1}{1 + C_d\left(\frac{1}{C_l} + \frac{1}{C_{ox}}\right)}\right] \tag{3.41}$$

$$= \frac{Q_s}{C_l}\left(\frac{\frac{1}{C_d}}{\frac{1}{C_d} + \frac{1}{C_l} + \frac{1}{C_{ox}}}\right) \tag{3.42}$$

The nonlinearity in this transduction comes from the depletion capacitance C_d. However, the condition $C_{ox} > C_l > C_d$ typically holds in practice, allowing the approximation

$$\frac{1}{C_l} + \frac{1}{C_{ox}} + \frac{1}{C_d} \approx \frac{1}{C_d} \tag{3.43}$$

and hence we find a quite linear transduction of charge to voltage:

$$\Delta V_{out} \approx \frac{Q_s}{C_l} \tag{3.44}$$

To give a feel for the numbers involved, a typical load capacitance C_l for this structure might be $\approx 40 - 80fF$, with a significant fraction of that due to the overlap capacitance C_{ol} from the adjacent phases. The resulting sensitivity given by Equation 3.44 would be on order $1-2\mu V/e-$.

3.5.1.11 The Buried Channel CCD

To increase transfer efficiency, the buried channel CCD was developed and is schematically shown in Figure 3-19. This structure is identical to the surface channel CCD except for the addition of a thin implanted n-layer beneath the CCD gates. Conceptually more difficult to understand, this structure confines charge a distance away from the semiconductor oxide interface, reducing the effects of interface traps. Figure 3-20 shows the energy band diagram for a buried-channel CCD. The pn junction formed between the n-layer and the p-substrate is reverse biased and the n region is depleted of mobile carriers. A potential maximum occurs some distance away from the oxide interface as shown in Figure 3-21, and this is where the signal charge lies, confined in all directions by reverse biased space charge regions. The analysis of the buried channel CCD is considerably more complicated than the surface channel CCD as the gate voltage, potential, and signal charge no longer obey a simple relationship due to the presence of the n-layer and the spreading of the signal charge [37]. However, the potential maximum, called the *channel potential* serves the same function as the surface potential in the surface channel CCD and a fairly linear variation with channel charge and gate voltage is observed [38]. Moving the gate voltage applied to the CCD gates shifts the channel potential, and this allows the continued usage of the water and well analogy.

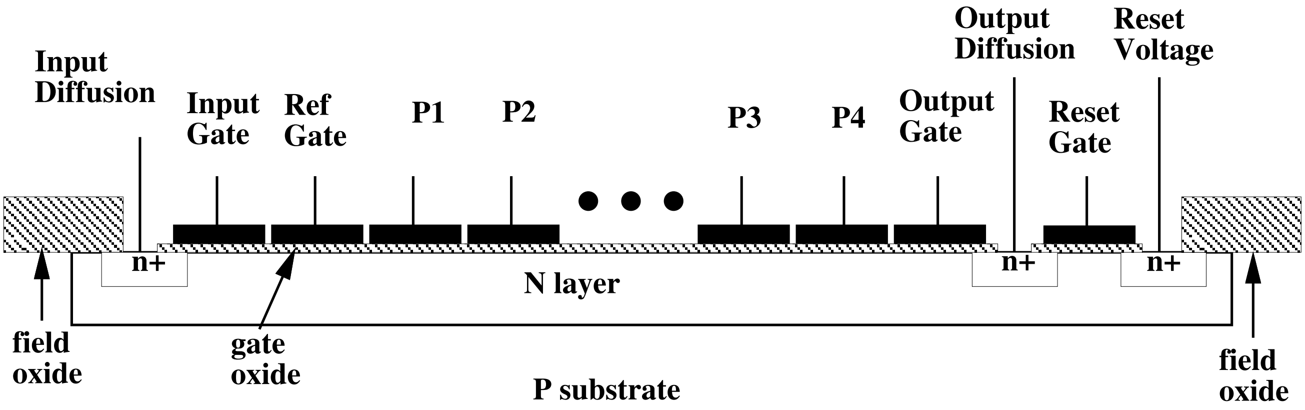


FIGURE 3-19: Diagrammatic representation of a typical buried-channel CCD, including input and output structures.

There are two main advantages to using buried channel CCDs: inefficiency goes down because it is easier to have far fewer traps in the bulk than at the surface, and the lateral fringing fields responsible for the intermediate stages of charge transfer are enhanced due to the increased distance between the signal charge and the gate as shown in Figure 3-22. With modern buried channel CCDs, transfer inefficiency is typically as small as 1×10^{-5} giving five nines of efficiency 0.99999. As a comparison, our one thousand transfers now results in only 1% percent of signal loss as opposed to 10% with the four nines of surface channel CCDs. However, the main drawbacks of buried channel CCDs are increased dark current and reduced charge capacity. An approximate relation between the maximum charge packet size for a surface channel CCD and a buried channel CCD given the same gate voltages and oxide thickness is derived in [37]:

$$\frac{Q_{max}(SCCD)}{Q_{max}(BCCD)} \approx 1 + \frac{\epsilon_{ox} x_n}{2\epsilon_{si} t_{ox}} \tag{3.45}$$

where x_n is the thickness of the implanted n-layer and t_{ox} is the thickness of the gate oxide. Assuming typical values of $t_{ox} = 0.04\mu m$ and $x_n = 0.5\mu m$, this ratio is about 3, indicating that the buried channel CCD has one third of the charge handling capacity of the surface device under the same conditions. As mentioned in 3.5.1.5, an example $100\mu m^2$ surface device has a capacity of $\approx 2.5Me-$, while a similar buried structure would have only $\approx 0.83Me-$. However, due to the vastly improved transfer efficiency of the buried device, it has become the workhorse of the charge-coupled device family.

3.5.2 The Interline CCD Imager with Storage

There are two main types of CCD array imagers: the *full frame transfer* and the *interline* imager [38]. In the full frame transfer array, there is an optically exposed area for image

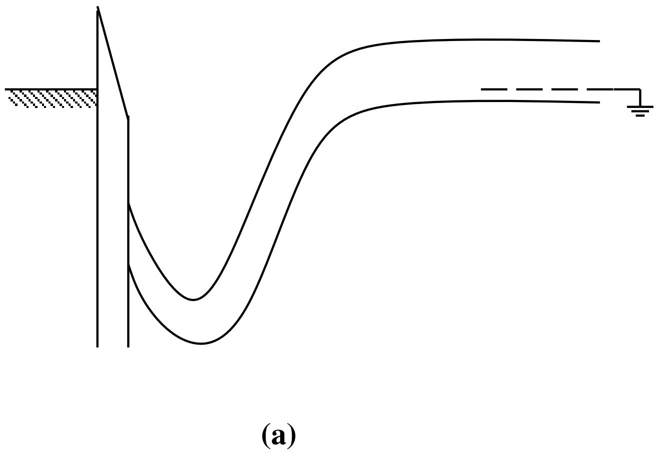


FIGURE 3-20: Energy band diagram for buried channel CCD with (a) no charge, and (b) with charge.

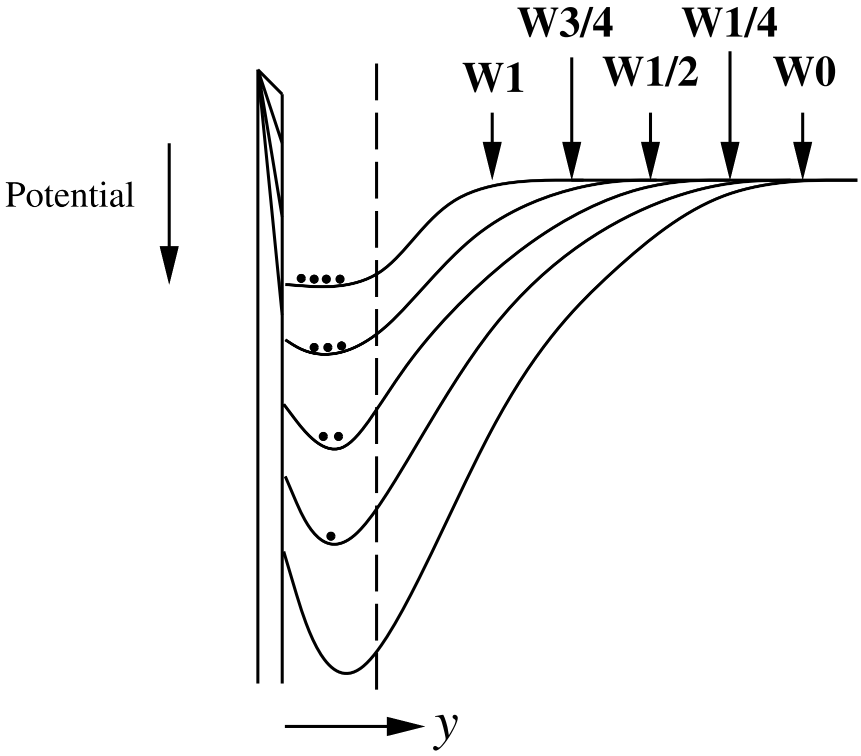


FIGURE 3-21: The potential into the semiconductor for a buried channel CCD with 0, 1/4, 1/2, 3/4, and full charge. After [37].

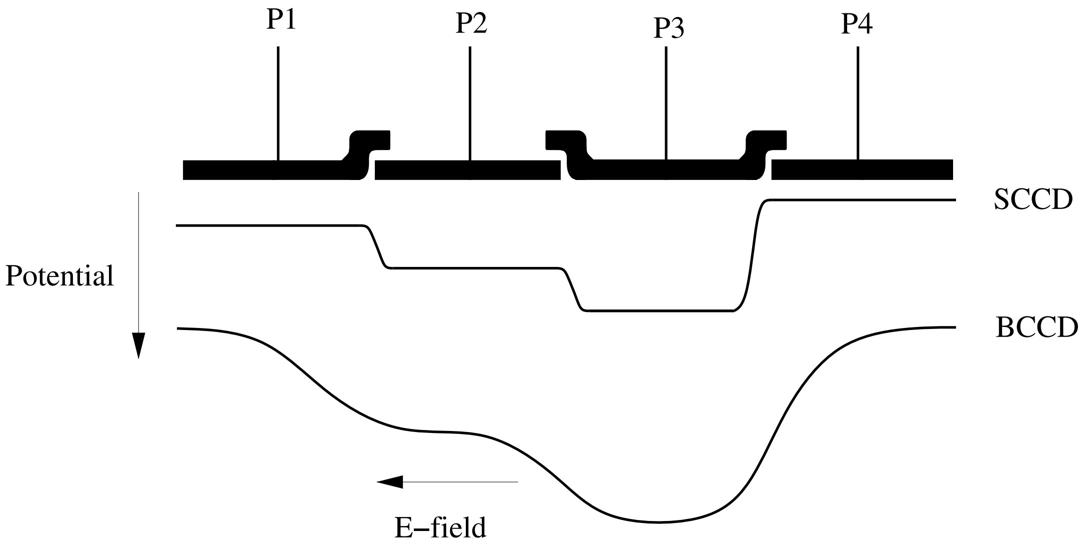


FIGURE 3-22: Burying the channel leads to smoother fringing fields enhancing transfer efficiency.

acquisition, and an area covered by a light shield. After acquisition, the entire image frame is transferred into the protected area, and then is typically read out serially while another image is acquired. The main advantage of this technique is that the fill factor (how much of the pixel area is devoted to gathering the charge generated by photons) of the im-ager can approach 100%. In the FOE chip, we require not one but two images in order to compute the brightness gradients using first centered differencing as shown in Figure 2-7. As we process data a column at a time, we need access to both images. This is difficult to provide efficiently using a full-frame approach, as this requires that the two images be interleaved. It was decided to use the second type of imager structure, the interline imager as shown in Figure 3-23, since it can quite easily be made to provide the interleaving of images.

In the interline topology, we have covered shift registers running alongside the optically exposed photo-gates. Once an image has been acquired in the photo-gates, it is typically shifted into the interline shift registers and then strobed out while a new image is acquired. In our application, we merely place two stages of shift register per pixel; this provides the additional storage required to hold the first image while the second image is acquired. After acquisition of the first image is complete, it is placed in the interline shift registers and shifted over once in the register to the right. After the acquisition of the second image is done, it is also placed into the register and then the image pair is strobed out together. The overall size of the embedded imager, taking up 90% of the chip area, is $6.9mm \times 6.9mm$. The actual pixel size including the two stages of interline register is $108\mu m \times 108\mu m$, and this was driven by the pitch in the processing array downstream. This size is quite a deal larger than commercial imaging chips, which have pixels that are typically an order of magnitude smaller in each dimension. These chips also have 100 times more picture elements than the $64 \times 64 = 4096$ pixels on the FOE

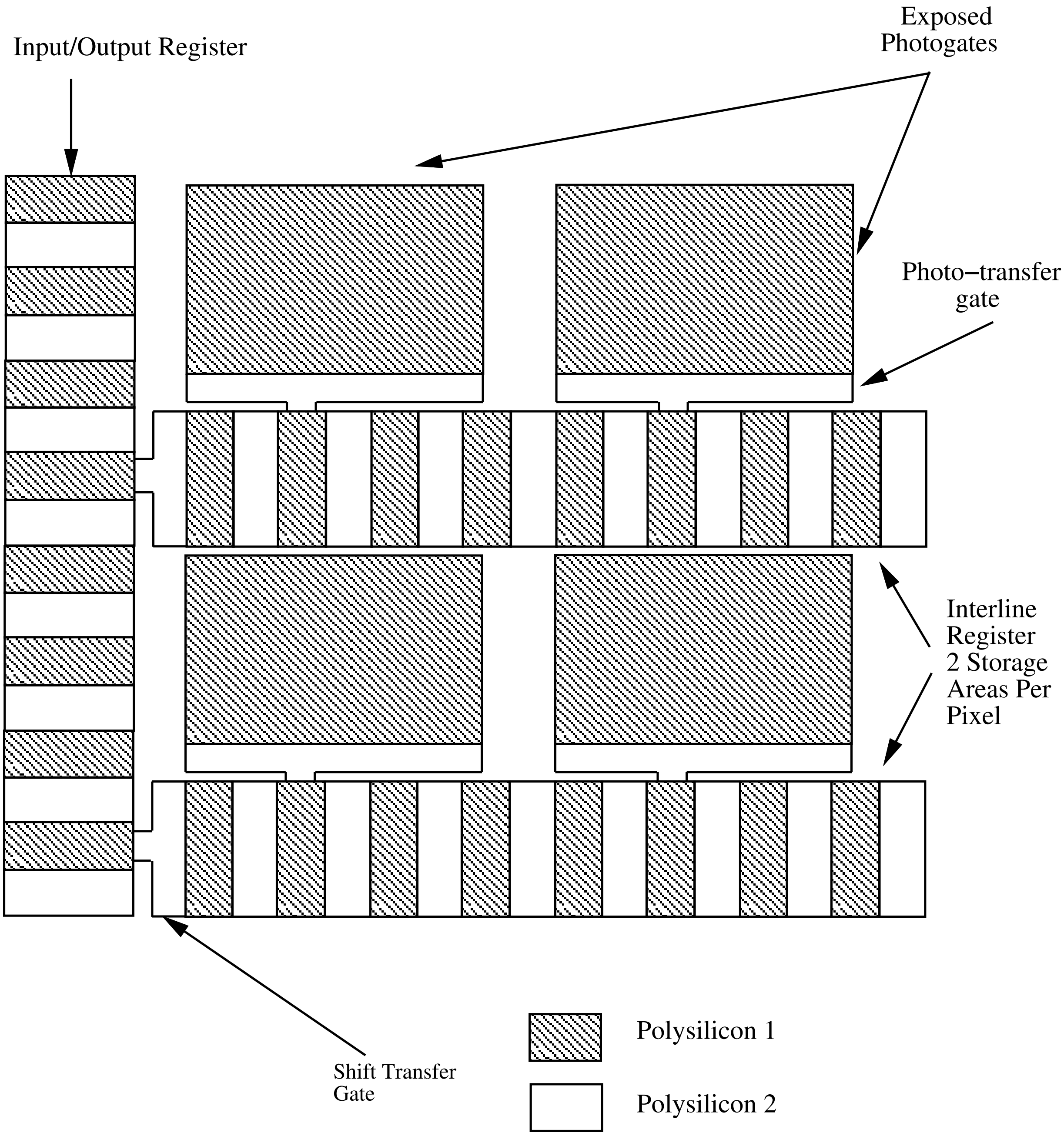


Figure 3-23: Schematic representation of the interline CCD imager on the FOE chip.

chip. For example, the Sony ICX022AL-3 interline CCD imager has pixel sizes of $11\mu m \times 13\mu m$ with $768 \times 493 = 0.38M$ pixels in an active area of $8.8mm \times 6.6mm$. Hence, the FOE chip collects signal over a comparable area, and each FOE pixel is the equivalent of ≈ 100 pixels of a standard CCD imager, and hence can acquire the same size charge packet ≈ 100 times faster as a result. This is what allows us to have vastly larger frame rates than standard cameras (30Hz), into the kilohertz range.

To the left of the imaging array is shown an input/output shift register running up the side where two stages per interline row are required to match the imager's pitch. This allows us to input data into the array and also take off-chip raw image data. Getting raw image data off of the chip is essential not only to quantify ideal algorithm performance, but also for calibration of the imaging parameters necessary to map the chip's 3-D motion to the resultant FOE location. Since the I/O register was intended for testing and calibration only, it was broken up into eight separate registers, each of which has a split-gate input structure on one end and a floating gate output structure at the other end. This was done to relax the requirements on the speed of output registers.

To the right of the imager, we have an array of floating gate amplifiers, with four per row as shown in Figure 3-24. Of the four outputs in a row, two are from two columns in the first image and other two are from the same column in the second image. The four values from these amplifiers, along with the four from the row above, provide the eight inputs needed for the analog processors in the CMOS array to estimate the brightness gradient.

Since the process that we will be using for fabrication of the FOE chip is buried channel, we need an appropriate lumped circuit model for the resulting floating gate output structure. This model is shown in Figure 3-25. Here we have a depletion capacitance C_{d2} from the channel to the substrate, as well as another depletion capacitance C_{d1} from the channel to the oxide. From the oxide to the gate we once again have C_{ox}, and loading the output gate we have $C_l = 2C_{ol} + C_i$ as before. The C_d that we had in the surface channel model is now split in two, with the injection point occurring in between. Notably, the analysis is the same as before except that C_d is replaced by C_{d2} while C_{ox} is replaced by the series combination of C_{d1} and C_{ox}. This results in the relation:

$$\Delta V_{out} = \frac{Q_s}{C_l}\left(\frac{\frac{1}{C_{d2}}}{\frac{1}{C_{d1}} + \frac{1}{C_{d2}} + \frac{1}{C_l} + \frac{1}{C_{ox}}}\right) \tag{3.46}$$

Now, we typically have the inequality $C_{ox} > C_l > C_{d1} > C_{d2}$. We can once again approximate to find:

$$\Delta V_{out} \approx \frac{Q_s}{C_l}\left(\frac{\frac{1}{C_{d2}}}{\frac{1}{C_{d1}} + \frac{1}{C_{d2}}}\right) = \frac{Q_s}{C_l}\left(\frac{C_{d1}}{C_{d1} + C_{d2}}\right) \tag{3.47}$$

Even though $C_{d1} > C_{d2}$, we leave their terms in place as C_{d2} may be as much as 30% of C_{d1} leading to a gain reduction of perhaps 25%. Typical sensitivity for this structure is on order $0.75 - 1.5\mu V/e-$. This equation would seem to imply a fair amount of nonlinearity due to the

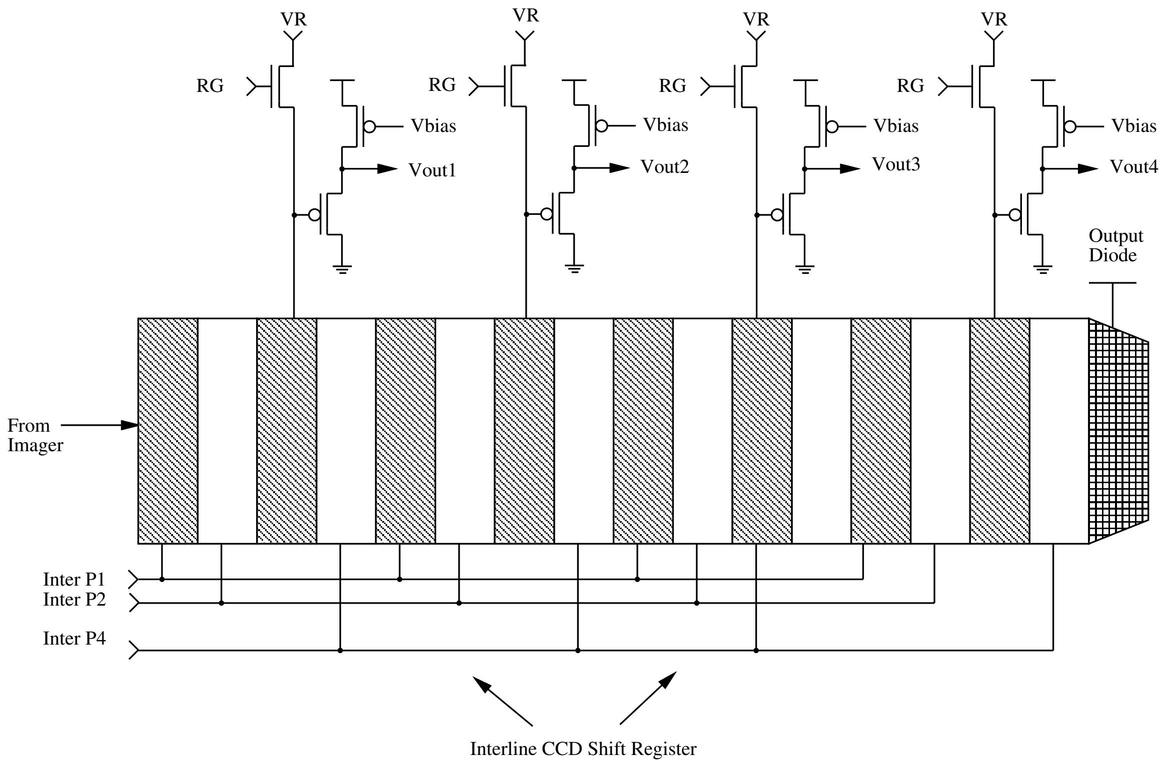


FIGURE 3-24: Estimating the image brightness gradients requires four floating gate outputs in a row.

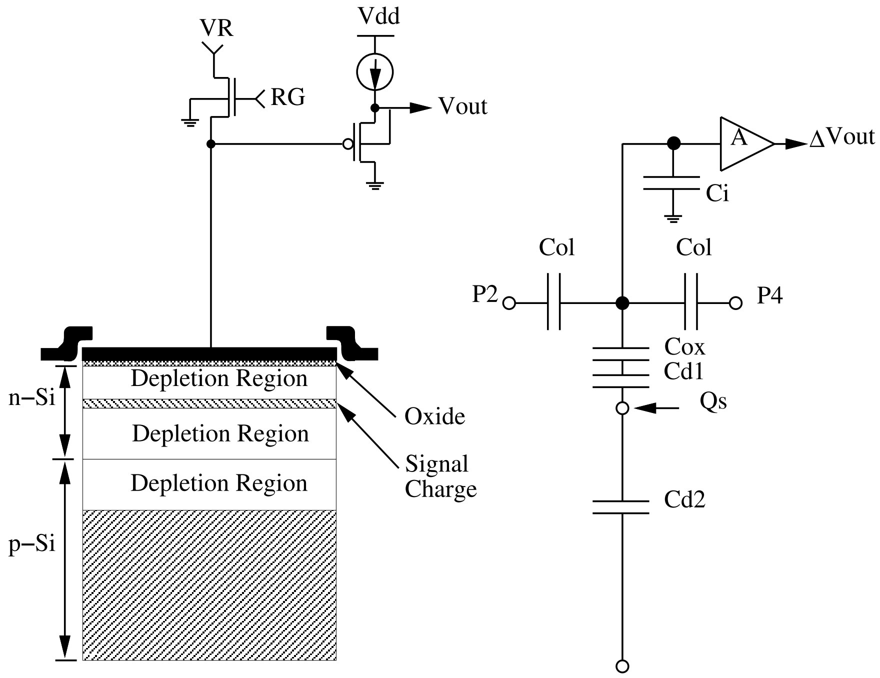


FIGURE 3-25: Lumped circuit model for the floating gate output structure in a buried channel CCD .

variation of the depletion capacitances; in practice the nonlinearities of C_{d1} and C_{d2} tend to be similar in nature, and the transduction is still quite linear [7].

3.5.3 The CMOS Processing Array

The CMOS processing array is a column of analog signal processors each of which is required to compute the following quantities in current which are then summed in current up the column and sent off-chip:

$$\begin{aligned} s_{abs} &= |E_t| \\ s_{quad} &= W(E_t,\eta)\Big(E_x^2 + E_y^2\Big) \\ e_x &= W(E_t,\eta)E_x\Big((x-x_0)E_x + (y-y_0)E_y\Big) \\ e_y &= W(E_t,\eta)E_y\Big((x-x_0)E_x + (y-y_0)E_y\Big) \end{aligned} \tag{3.48}$$

where we will use the cutoff weighting function:

$$W(E_t,\eta) = \begin{cases} 1 & \text{if } |E_t| < \eta \\ 0 & \text{otherwise} \end{cases} \tag{3.49}$$

We now present a brief discussion of the functioning of the processors designed to compute these quantities, after which we will describe the circuit structures in the implementation in detail. Figure 3-26 shows a block diagram representation of the signal flow of the processor, where we have augmented the circuit to include a row masking bit to facilitate testing. A digital shift register runs up the side of the processing array, and we will be able to shift in a computational mask. The idea is to be able to selectively turn off processors and prevent them from contributing to the sum. This way we can probe individual processors, as well as mask out during normal operation any that are substantially defective.

Using the eight voltages from the floating gate sense amplifiers, voltage in/current out transconductors are used, along with the appropriate current mirroring, to implement the linear combinations necessary to form the image brightness gradients E_x, E_y, and E_t. A current mode absolute value circuit forms $|E_t|$. This is used by the weighting function, which consists of a current difference with a reference current I_η followed by a latch. The mask bit, if asserted from the mask register, sets this latch so that the weighting function always evaluates low, accomplishing the desired masking. A copy of $|E_t|$ is made, again using mirrors, and sent off-chip through an in-line pass transistor controlled by the mask bit, giving the first output s_{abs}. The result of the comparison from the latch in the weighting function is used on pass gates placed in-line with the currents E_x and E_y; these signals are then used for the rest of the computation and this accomplishes the binary weighting of the cutoff function. To generate the second output s_{quad}, we take the now weighted gradient currents and pass each of them through a current mode squarer. The output of the squarers are then summed in current, sent through a pass transistor controlled by the masking bit, and from thence off-chip as s_{quad}.

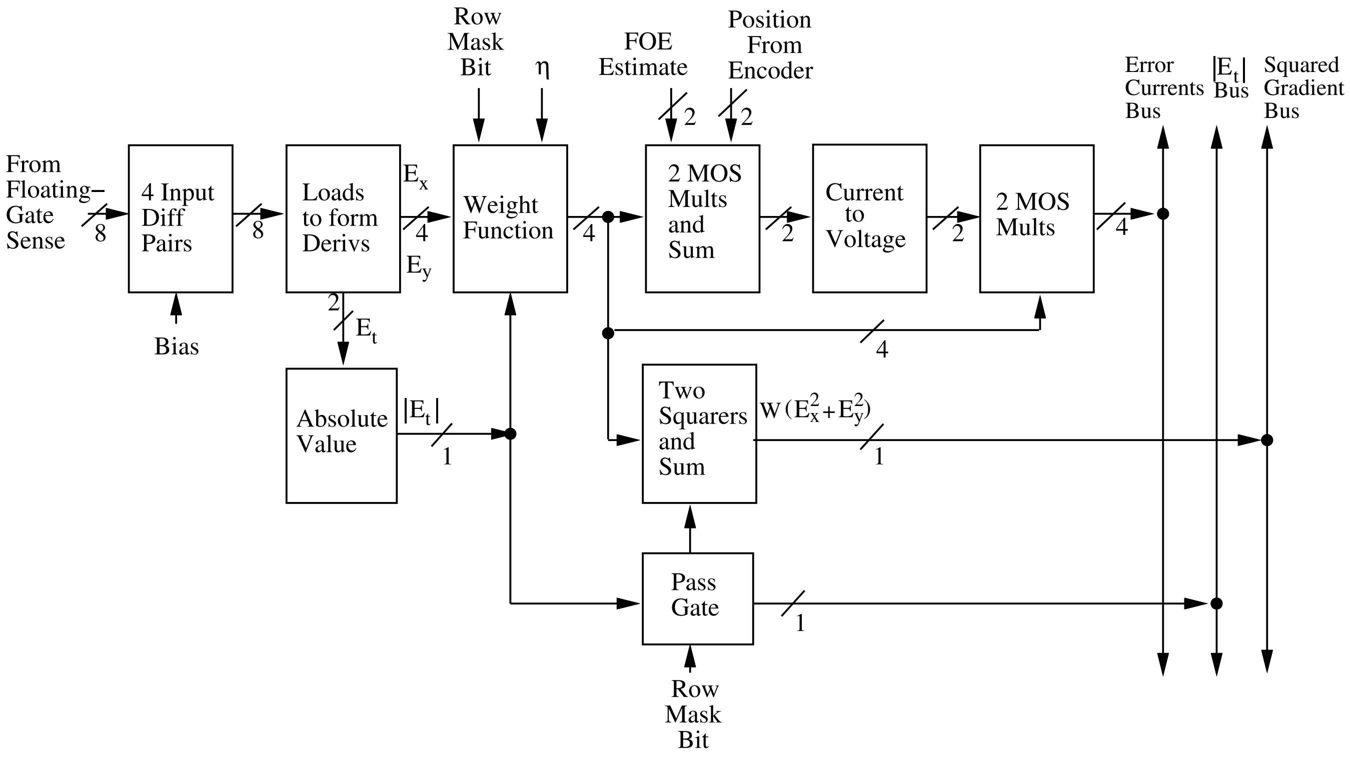


FIGURE 3-26: Block diagram indicating the structure of analog row processor in the CMOS processing array.

To form the error outputs e_x and e_y, copies of the weighted gradient currents are sent to a core of four analog multipliers. These four-quadrant multipliers are all of the same type: one input is in voltage, the second input is in current, and the output is also in current. The multiplier core is divided into two layers of two multipliers. In the first layer, differential voltages representing $x - x_0, y - y_0$ are one input to the multipliers, with the weighted gradient currents serving as the other input. The output currents from this first layer of multipliers are summed, forming the dot product $(x - x_0)E_x + (y - y_0)E_y$. In the second layer of multipliers, we need to multiply this dot product with the weighted image gradients. Since both signals are now in current, we transduce the dot product signal into a voltage using a current to voltage converter. This allows us to use the same multipliers in the second layer as in the first. The outputs of this second layer are the desired error currents; they are summed up the column by KCL and then sent off-chip.

3.5.3.1 Estimating the Brightness Gradients

The first stage of the CMOS processor is designed to estimate the brightness gradients E_x, E_y, and E_t from the eight pixel voltages provided by the floating-gate amplifier array. By rearranging the estimators for the gradients, we can reduce the computational load substantially. If we define the corner differences in our $2 \times 2 \times 2$ cube of pixels,

$$\begin{aligned}
\delta_1 &= E(i+1,j+1,k+1) - E(i,j,k) \\
\delta_2 &= E(i+1,j,k+1) - E(i,j+1,k) \\
\delta_3 &= E(i+1,j,k) - E(i,j+1,k+1) \\
\delta_4 &= E(i+1,j+1,k) - E(i,j,k+1)
\end{aligned} \tag{3.50}$$

and refer to Equations 2.79, 2.80, and 2.81, we note that:

$$\begin{aligned}
E_x &\approx \frac{1}{4}(\delta_1 - \delta_2 - \delta_3 + \delta_4) \\
E_y &\approx \frac{1}{4}(\delta_1 + \delta_2 + \delta_3 + \delta_4) \\
E_t &\approx \frac{1}{4}(\delta_1 + \delta_2 - \delta_3 - \delta_4)
\end{aligned} \tag{3.51}$$

which shows that E_x, E_y, and E_t are mutually orthogonal linear combinations of δ_1, δ_2, δ_3, and δ_4. Hence, in order to form the gradients we need only four *differential* transconductors and by using current mirrors we can form the linear combinations of their outputs necessary to implement Equation 3.51.

There are many possible circuits to implement the necessary voltage to current transduction, many of which offer excellent range and linearity [47, 48, 49]. Unfortunately, the size constraints imposed by the level of integration we are attempting are formidable, and make the simplicity of the design paramount. Furthermore, the least-squares nature of the algorithm we are implementing allows us to use simpler, less accurate components. The circuit we use to do the transduction is the simple source coupled pair, as shown in Figure 3-27.

Using the simple quadratic model for the MOS transistor relating the gate to source voltage V_{gs} to the drain current I_d:

$$I_d = \mu \frac{C_{ox} W}{2L} (V_{gs} - V_t)^2 \tag{3.52}$$

we can derive the DC transfer characteristic of the source coupled pair [50]:

$$\Delta I_{out} = \mu \frac{C_{ox} W}{2L} (\Delta V_{in}) \sqrt{\left(\frac{2I}{\mu (C_{ox} W / 2L)} \right) - (\Delta V_{in})^2} \tag{3.53}$$

This expression is only valid when both transistors are in saturation. To satisfy this requirement, we must have

$$|\Delta V_{in}| \le V_r = \sqrt{\frac{I}{\mu (C_{ox} W / 2L)}} \tag{3.54}$$

Normalizing this equation, we can define:

$$\beta \equiv \frac{\Delta I_{out}}{I}, \quad \eta \equiv \frac{\Delta V_{in}}{V_r} \tag{3.55}$$

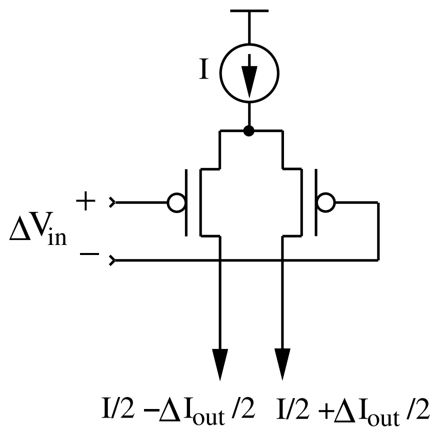


FIGURE 3-27: A simple source coupled pair is used to transduce voltage signals from the floating gate outputs to current signals for gradient estimation.

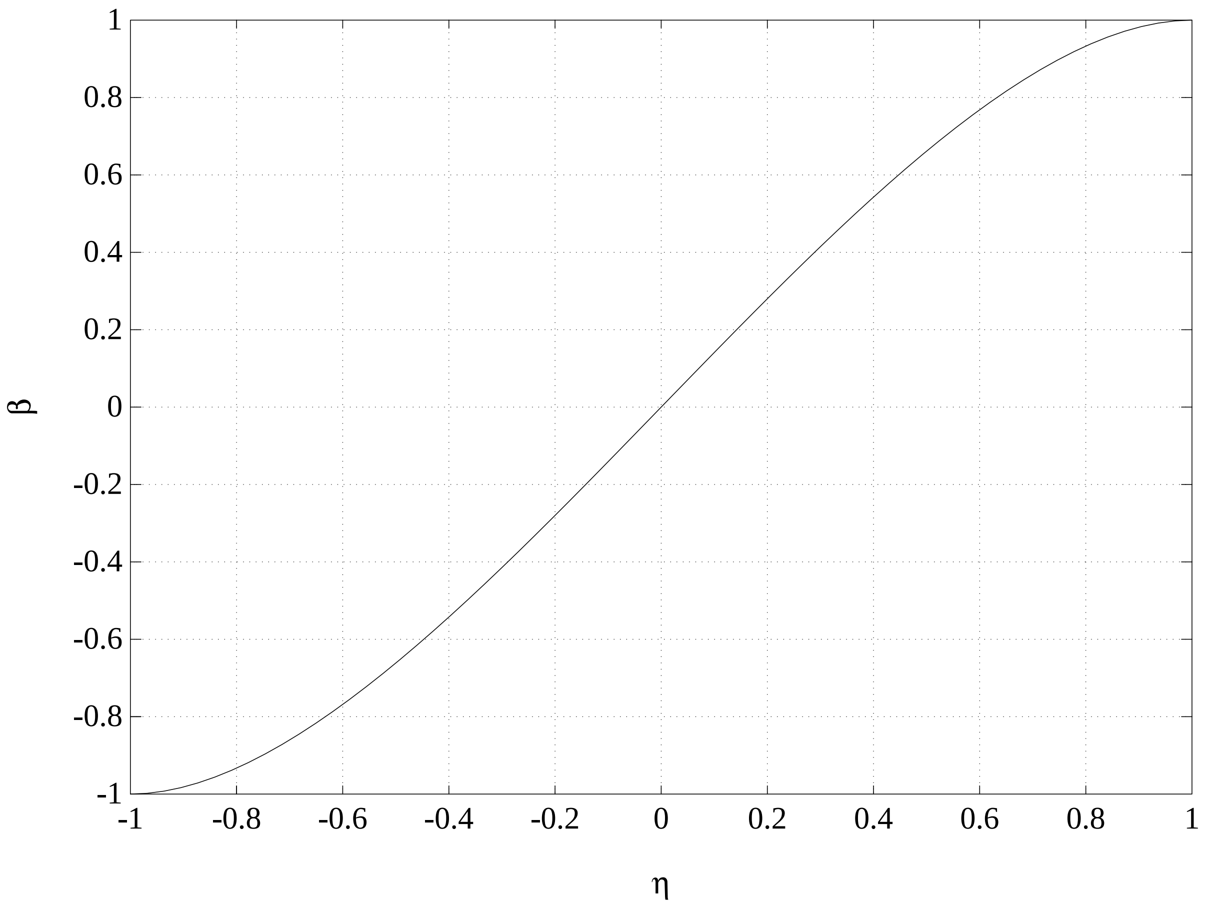


FIGURE 3-28: Normalized input differential voltage to output differential current transfer characteristic of the source-coupled pair.

which results in:

$$\beta = \begin{cases} -1 & \text{if } \eta \leq -1 \\ \eta\sqrt{2-\eta^2} & \text{if } \eta^2 < 1 \\ 1 & \text{if } \eta \geq 1 \end{cases} \tag{3.56}$$

A graph of this function is shown in Figure 3-28. Near the origin, the curve appears quite linear, with a normalized transfer characteristic of

$$\beta \approx \sqrt{2}\eta \tag{3.57}$$

The required four input transconductors for brightness gradient estimation are shown in Figure 3-29 along with the appropriate cascoded mirroring to form the desired linear combinations of Equation 3.51. Recall that the input differential voltages on these PMOS source coupled pairs come from the outputs of the floating gate sense amplifiers from the CCD section. Using the approximate numbers of $Q_{max} = 1Me-$ with floating gate sensitivities of $1\mu V/e-$, we expect input signal swings to be on order $1V$. Hence, we can set the input range of the source-coupled pairs V_r to match the output signal swing from the floating gate amplifiers.

Since we use an n-well process, we can tie the back gates of PMOS devices to their respective sources. This eliminates the back-gate effect, which causes the threshold voltage of the devices to vary with V_{sb}, the source to back-gate voltage. Additionally, due to the reduced mobility of holes compared with electrons, PMOS devices are 2–3 times more resistive than NMOS devices for the same channel length. This means that, with the same gate geometry and bias current, a PMOS channel source coupled pair will have a larger active range than an NMOS source coupled pair. Thus, we use PMOS source coupled pairs. For the Orbit process, $\kappa = \mu C_{ox}/2 \approx 25\mu A/V^2$ for n-channel devices and $\approx 10\mu A/V^2$ for p-channel devices. With a current of $I = 5\mu A$, this requires the PMOS devices in the source-coupled pair to have $W/L = 0.5 = 6\mu m/12\mu m$. On the chip, the biasing current sources I are implemented using standard improved cascode current sources [50].

3.5.3.2 The Cutoff Weighting Function

Now that we have differential currents $\Delta I_x, \Delta I_y$, and ΔI_t representing the brightness gradients E_x, E_y, and E_t we need to implement the weighting function and use its decision to gate ΔI_x and ΔI_y to the rest of the processor. Recall that the cutoff weighting function performs the following operation:

$$W(E_t, \eta) = \begin{cases} 1 & \text{if } |E_t| < \eta \\ 0 & \text{otherwise} \end{cases} \tag{3.58}$$

Figure 3-30 demonstrates a block diagram for the circuit which was designed to perform this computation. We first take the absolute value of the differential current representing E_t, forming a single ended current $|2\Delta I_t|$. The difference between this current and a reference current $2I_\eta$ is then sensed by a current comparator resulting in the signal $\bar{W}$. Additionally, an extra copy

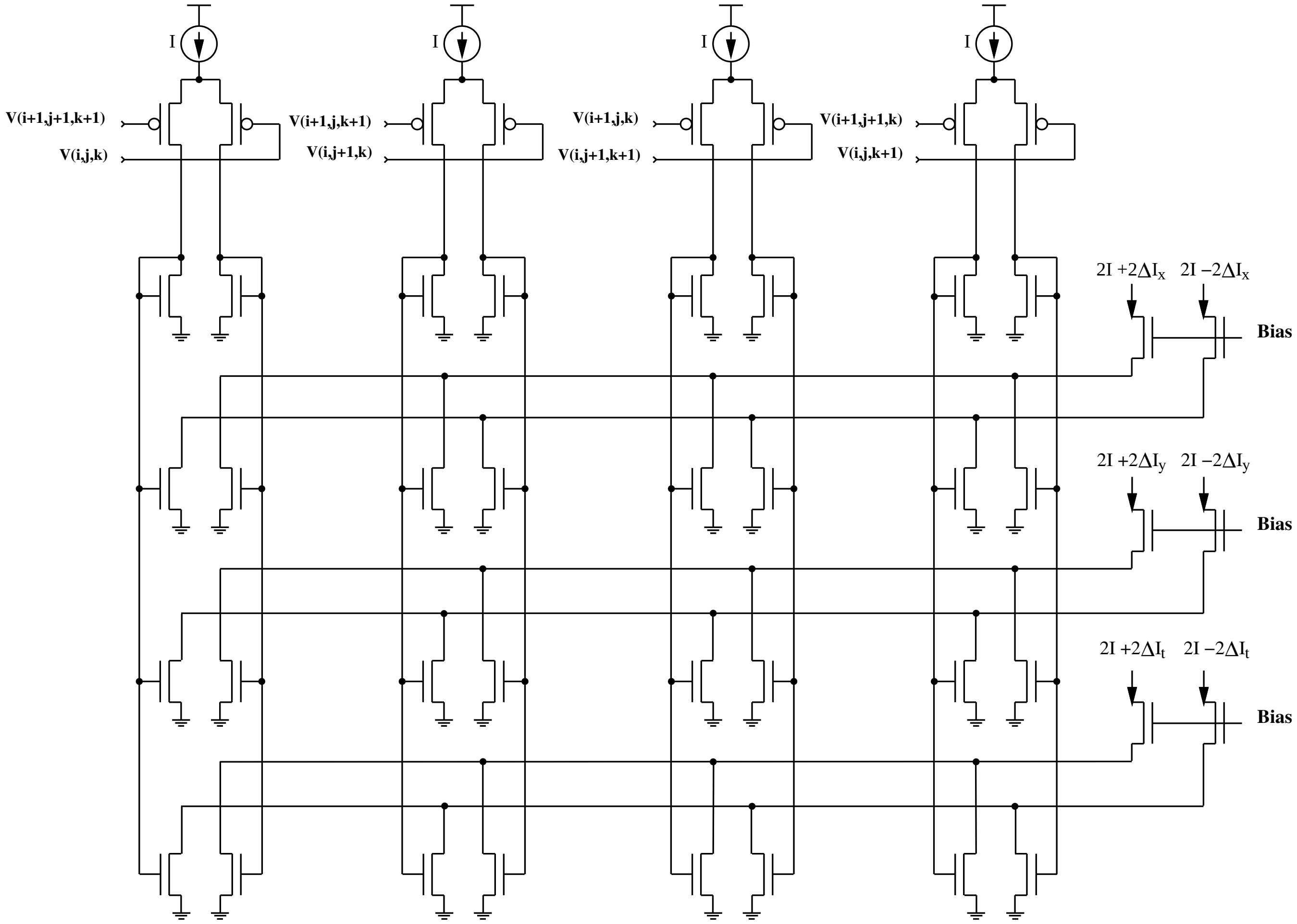


Figure 3-29: Using transconductors and mirrors to generate the first centered difference approximations to the image gradients.

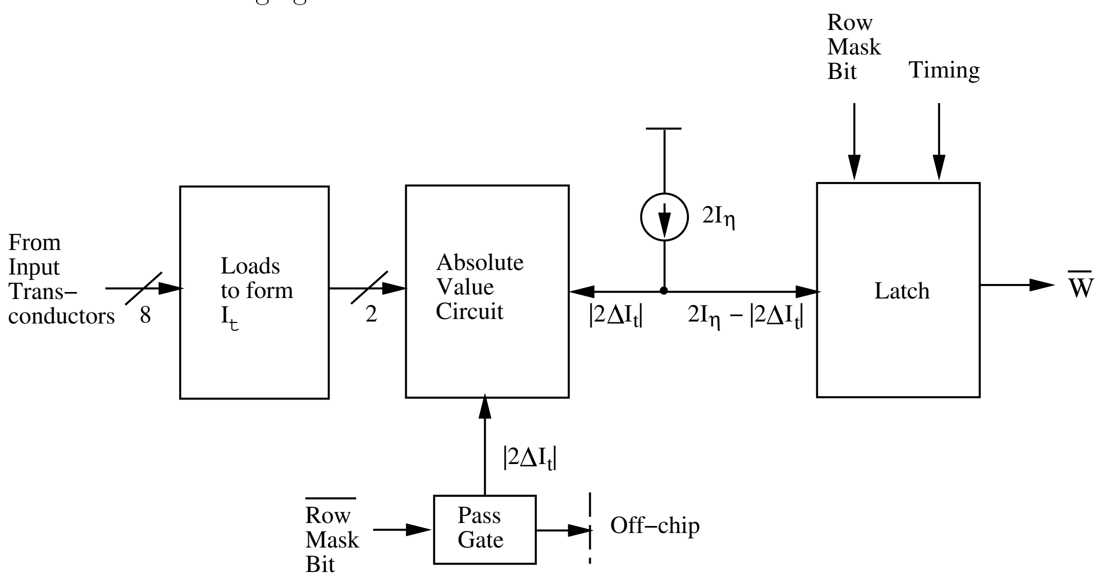


Figure 3-30: Block diagram representation of the Cutoff Weighting Function.

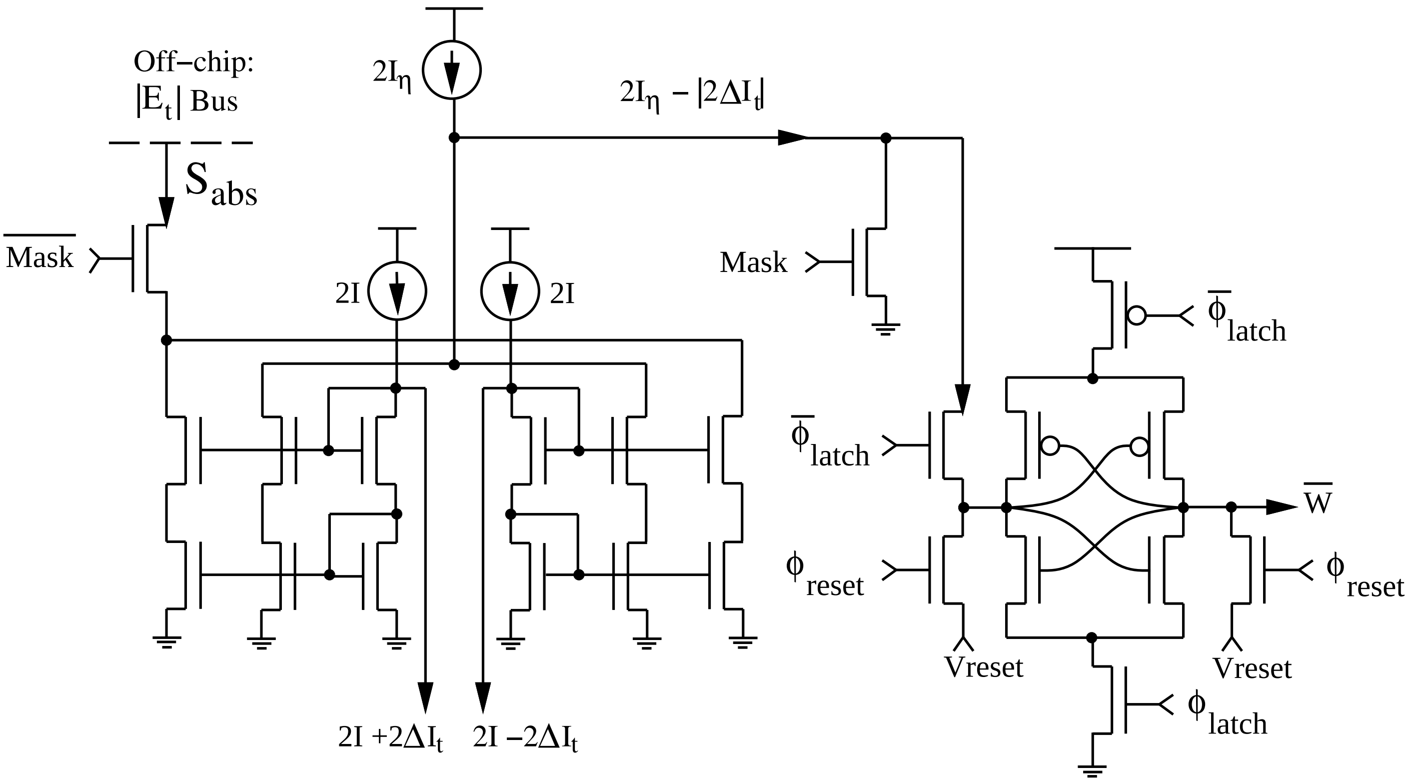


FIGURE 3-31: Circuit implementation of the Cutoff Weighting Function.

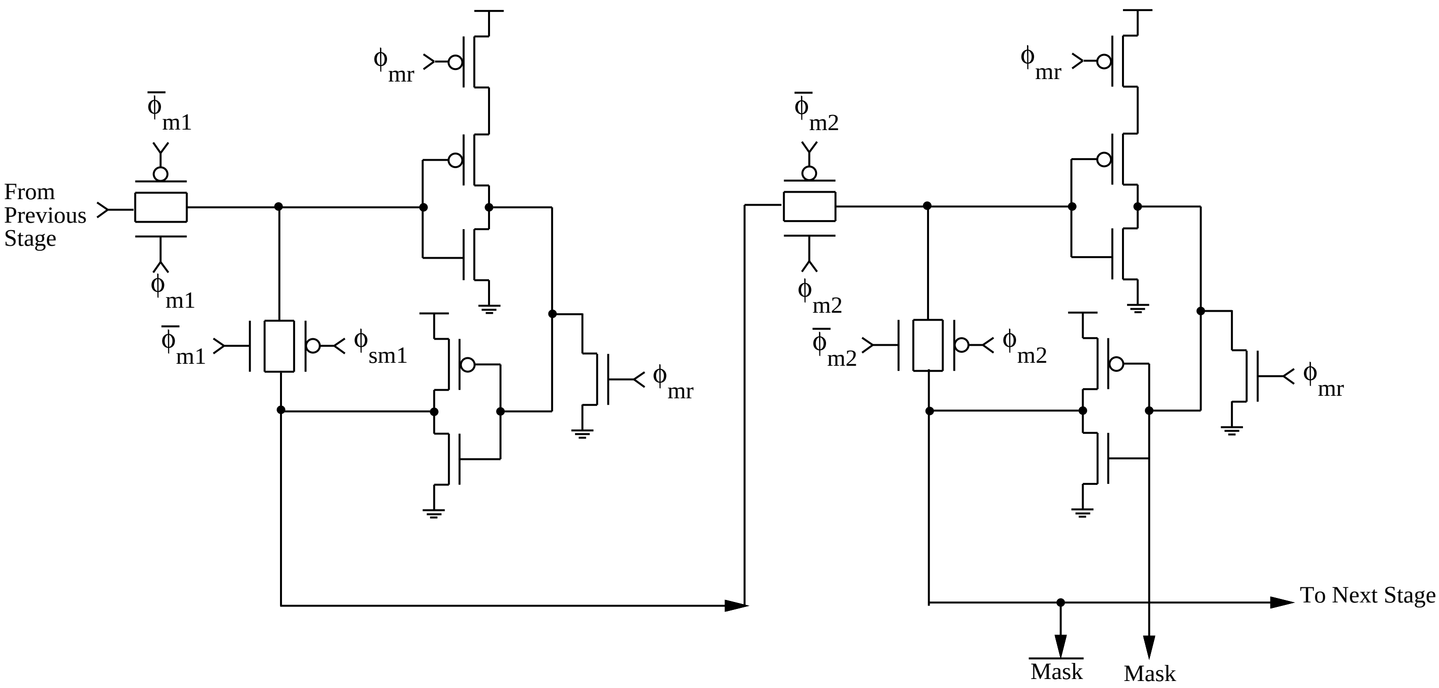


FIGURE 3-32: One stage of the masking shift register.

of the $|2\Delta I_t|$ signal is made and sent off-chip through a mask-bit controlled pass gate. The full circuit implementation for the cutoff weighting function is shown in Figure 3-31.

The E_t current ΔI_t is differential and balanced, so we can represent it by:

$$\begin{aligned} I_+ &= 2I + 2\Delta I_t \\ I_- &= 2I - 2\Delta I_t \end{aligned} \tag{3.59}$$

The two current sources of size $2I$ subtract the bias current off of I_+ and I_-. The remaining current flows through the double diode-connected NMOS transistors if it is positive. Thus, if $\Delta I_t > 0$, the right double diode will carry the current $2\Delta I_t$ while the left double diode will carry no current. Similarly if $\Delta I_t < 0$, the right double diode will carry no current and the left double diode will have $-2\Delta I_t$. We mirror off these two currents and add them together. The resulting current is $|2\Delta I_t|$, as desired. An extra copy of $|2\Delta I_t|$ is easily obtained by duplicating the current mirrors; this signal is sent through a pass transistor controlled by the complement of the mask bit and from there off-chip for the absolute value channel.

We now subtract this $|2\Delta I_t|$ current from a reference current $2I_\eta$ as shown. If the resulting current is positive, then $|E_t| < \eta$ and $\bar{W}$ should be low. If the resulting current is negative, then $|E_t| > \eta$ and $\bar{W}$ should be high. We feed this current difference into the latch at the right of the diagram. While the reset signal ϕ_{reset} is high, the reset transistors pre-charge the nodes of the latch to a voltage V_{reset}. During this time, the signal current flows into the latch and out of the left reset transistor. When the reset signal goes low, the comparator input current is integrated onto the left parasitic capacitor of the latch, forming a voltage difference across the latch. When the latch signal ϕ_{latch} goes high, the latch amplifies the voltage difference to the rails. Since this comparator is not driven differentially it will exhibit offset. However, this merely changes the effective value of η. To minimize this offset, the complimentary signals ϕ_{latch} and $\bar{\phi}_{latch}$ can be driven with controllable rise and fall times.

A pull-down at the input to the latch and driven by the masking bit is provided. When this bit is asserted, one side of the latch is always pulled down, and the latch evaluation will always result in $\bar{W} = 1$, effectively masking out the processor. One stage of the masking shift register used to shift in the mask bits is shown in Figure 3-32. It is a simple modification of a standard static latch based structure, employing 2-phase non-overlapping clocking [51]. It is composed of two identical sub-blocks, each one driven by one of the phases, and each of these sub-blocks consists of an inverter pair with some pass gates. When the appropriate clock phase is not active, the two inverters are closed in a feedback loop. When the phase is active, the feedback loop is broken and the input from the previous substage is driven in. The reset transistors force a logic 0 into the inverter pair when ϕ_{reset} is asserted. This has the effect of setting all the mask bits up and down the register to zero, indicating that no processors are masked out in the computation. This is intended to be the normal mode of operation.

Now that the cutoff weighting decision has been made, we need to apply it to the image gradients. The circuit for doing so is shown in Figure 3-33. In this circuit structure, we take the currents $\Delta I_x, \Delta I_y$, pass them through a pass transistor controlled by the weighting function decision, and into a p-channel cascoded current mirror sized 2:1. This accomplishes the weighting function, and we make three copies of the resultant binary weighted currents. One copy is used for the squared gradient magnitude, while the other two are used in the multiplier core.

3.5.3.3 Getting the Brightness Gradient Squared Magnitude

To compute the brightness gradient magnitude, we need a circuit whose characteristic provides a square. Since the MOS transistor is a square-law device, achieving an output current which is the square of an input voltage is a natural application [52]. Furthermore, one common approach to linearization of the source coupled pair that we have discussed is to add a quadratic term into the tail current [49]. However, we need a circuit which operates with current input. Certainly, we can use translinear circuits to accomplish such a function [47, 53]. These types of circuits are based on the logarithmic nature of the bipolar transistor, and indeed the Orbit process includes vertical *npn*s, albeit without the usual collector implant to reduce collector resistance. The collector resistance need not affect the performance of translinear circuits, so this is a viable option. However, the size of the available bipolar is substantial, and an MOS current-mode circuit is preferable. Such a circuit is described in [54] and is the topology we use on the FOE chip. The basic three transistor core is shown in Figure 3-34.

Clearly, the bias voltage V_b set up by the bias current I_0 in the two transistor bias tree is

$$V_b = 2V_t + 2\Delta V \tag{3.60}$$

where $I_0 = \kappa \Delta V^2$ and $\kappa = \mu C_{ox}/2$. Rearranging this, we see that:

$$I_0 = \frac{\kappa}{4}(V_b - 2V_t)^2 \tag{3.61}$$

Turning to I_1, we note that

$$\begin{aligned} I_1 &= \kappa(V_b - V_i - V_t)^2 \\ &= \kappa\left((V_b - 2V_t) - (V_i - V_t)\right)^2 \\ &= \kappa(V_b - 2V_t)^2 - 2\kappa(V_b - 2V_t)(V_i - V_t) + \kappa(V_i - V_t)^2 \end{aligned} \tag{3.62}$$

But the first term is just $4I_0$ and the last term is I_2. We can write the middle term as:

$$2\kappa(V_b - 2V_t)(V_i - V_t) = 2\kappa\left(\sqrt{\frac{4I_0}{\kappa}}\right)\left(\sqrt{\frac{I_2}{\kappa}}\right) = 4\sqrt{I_0 I_2} \tag{3.63}$$

Clearly then,

$$I_1 = 4I_0 - 4\sqrt{I_0 I_2} + I_2 \tag{3.64}$$

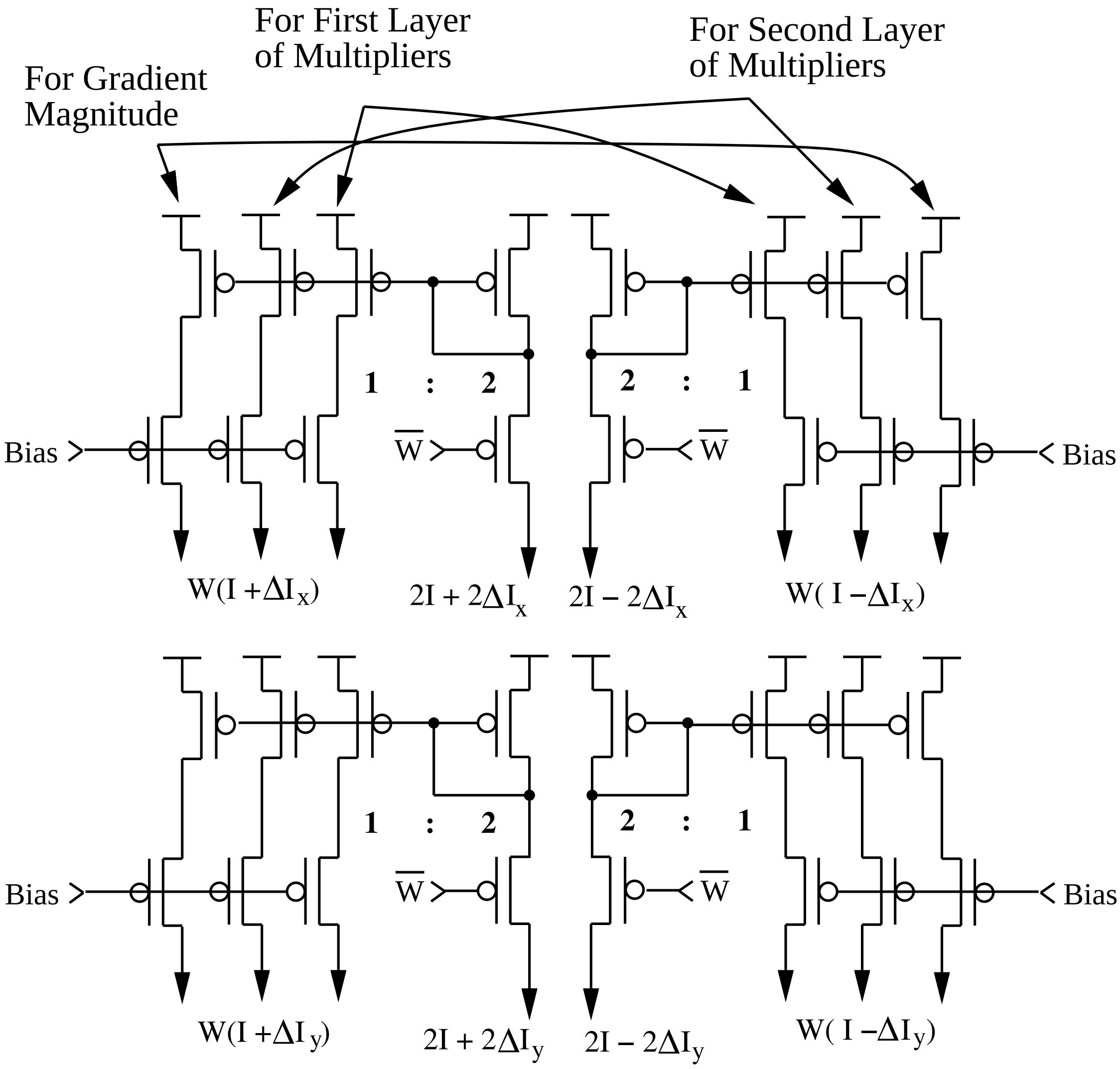


FIGURE 3-33: Applying the result of the Cutoff Weighting Function to the brightness gradient.

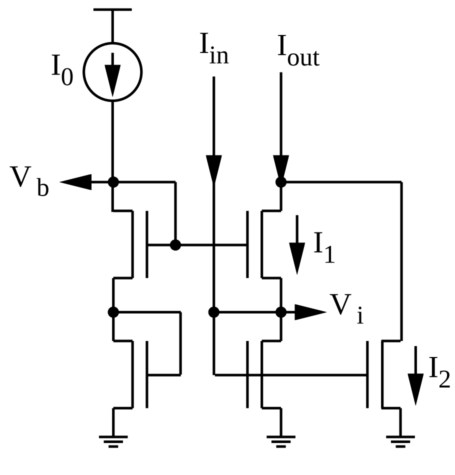


FIGURE 3-34: A simple three transistor current mode MOS quadratic circuit with its bias tree.

Rearranging to isolate the square root, squaring to eliminate it, and collecting terms results in:

$$(I_2 - I_1)^2 - 8(I_2 + I_1)I_0 + 16I_0^2 = 0 \tag{3.65}$$

But $I_{in} = I_2 - I_1$ and $I_{out} = I_2 + I_1$ and hence

$$I_{in}^2 - 8I_{out}I_0 + 16I_0^2 = 0 \tag{3.66}$$

and therefore

$$I_{out} = 2I_0 + \frac{I_{in}^2}{8I_0} \tag{3.67}$$

giving a one-quadrant current-mode squarer. To find the limits of operation of this squarer, we note that in order for the transistor carrying I_1 to be in the saturation region:

$$\begin{aligned} V_b - V_i > V_t \Rightarrow & \quad V_i < V_b - V_t \\ \Rightarrow & \quad V_i - V_t < V_b - 2V_t \\ \Rightarrow & \quad \kappa(V_i - V_t)^2 < \kappa(V_b - 2V_t)^2 \\ \Rightarrow & \quad I_{in} < 4I_0 \end{aligned} \tag{3.68}$$

We need a two quadrant version of this circuit, and a method of doing so is to take the each leg of the differential input current, square it separately, and sum the results. If $I_+ = I + \Delta I$ and $I_- = I - \Delta I$, then we have

$$\begin{aligned} I_{out} &= 2I_0 + \frac{(I + \Delta I)^2}{8I_0} + 2I_0 + \frac{(I - \Delta I)^2}{8I_0} \\ &= 4I_0 + \frac{I^2 + 2I\Delta I + \Delta I^2 + I^2 - 2I\Delta I + \Delta I^2}{8I_0} \end{aligned} \tag{3.69}$$

The cross-terms cancel, and we are left with:

$$I_{out} = 4I_0 + \frac{I^2 + \Delta I^2}{4I_0} \tag{3.70}$$

For proper operation of this circuit, we require that the common-mode current $2I < 4I_0$. Notice that the output current has an offset that is dependent on the common-mode current. This is not a difficulty as the common mode current is fixed in our gradient currents, so this is a *constant* offset and can be subtracted off. Once this offset has been cancelled, note that the maximum output signal current that we can draw during the proper operation of this differential squarer is $I/2$. Beyond this region the circuit merely mirrors the input current, and the I/O characteristic linearizes.

Figure 3-35 shows the full circuit implementation for calculating the squared gradient. Each leg of each differential current goes through a three transistor squarer with a cascoding transistor on top. The four resulting output currents are summed, a bias transistor removes the offset, and the result is mirrored off. This output is now:

$$s_{quad} = W\frac{\Delta I_x^2 + \Delta I_y^2}{4I_0} \tag{3.71}$$

and is summed up and down the array by KCL to form the quadratic channel output.

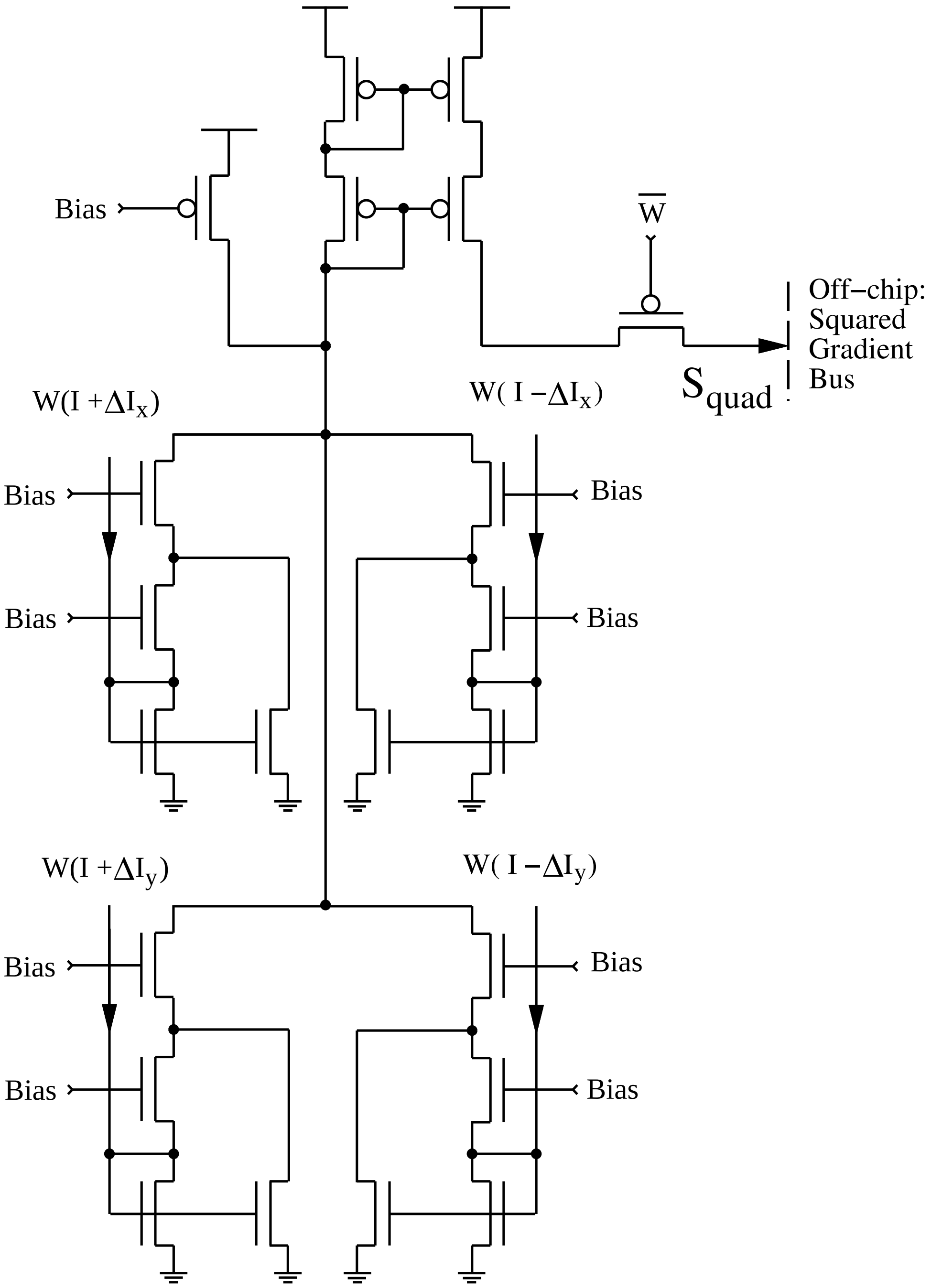


FIGURE 3-35: Full circuit implementation for finding the brightness gradient squared magnitude.

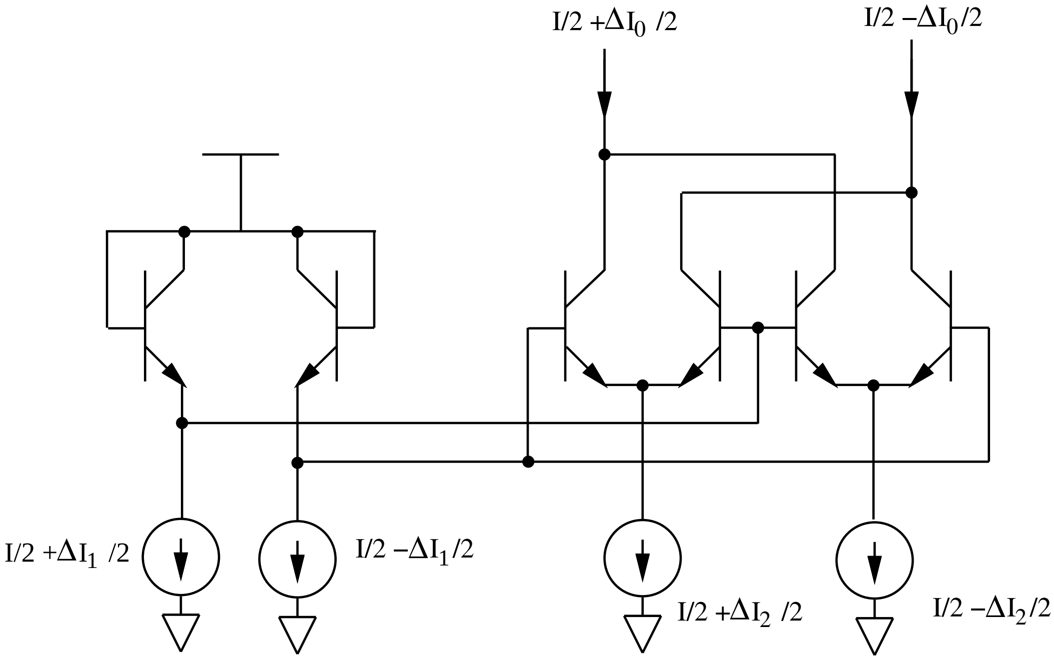


FIGURE 3-36: Four-quadrant bipolar Gilbert multiplier.

3.5.3.4 The Multiplier Core

Now we turn to the four multipliers at the core of the processor. These multipliers are required to have four quadrant operation (both inputs are signed), with one operand in differential voltage and the other in differential current. The four-quadrant bipolar multiplier based on the translinear principle and shown in Figure 3-36 has dominated the industry since its invention by Barrie Gilbert in the late 60s [55, 56]. The core of this topology is formed from six matched bipolar transistors. The four bipolars on the right have a DC transfer characteristic of [50]

$$\Delta I_{out} = \Delta I_2 \tanh\left(\frac{\Delta V}{2V_{th}}\right) \tag{3.72}$$

where ΔV is the voltage difference applied to the bases of the bipolars and V_{th} is the thermal voltage. The two bipolars on the left form a pre-distortion circuit. The voltage difference produced at the emitters of the two bipolars by the differential current ΔI_1 on the left is:

$$\Delta V = 2V_{th} \tanh^{-1}\left(\frac{\Delta I_1}{I}\right) \tag{3.73}$$

Combining these, we find an ideal four quadrant multiplication of the differential currents:

$$\Delta I_{out} = \frac{\Delta I_1 \Delta I_2}{I} \tag{3.74}$$

To have voltage input, one can generate the differential currents using transconductors. This bipolar structure is still very much the topology of choice [57, 53, 58, 59, 47], so much so that in many classic analog circuit texts it is required reading [50, 60]. Unfortunately, we cannot

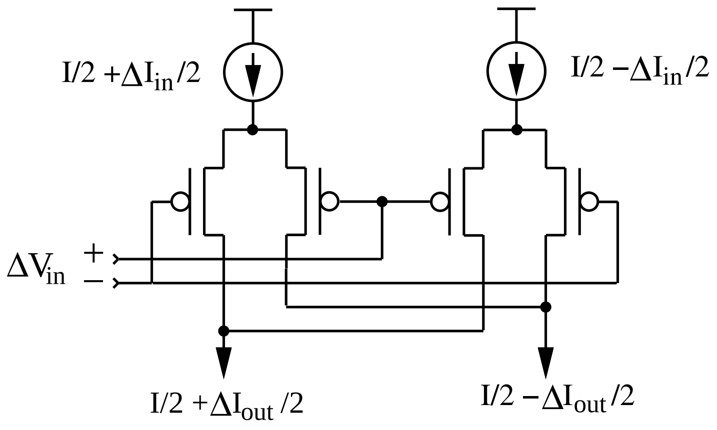


FIGURE 3-37: A simple MOS version of the Gilbert multiplier.

use bipolars in our design due to their size, so we need an MOS version. Much effort has been invested over the years in designing multipliers in MOS. These topologies fall into two basic categories: approaches based on the Gilbert multiplier [61, 62, 63] and approaches based on the so called quarter-square technique [64, 52, 65, 66]. In the approaches based on the Gilbert topology, the pre-distortion circuit is omitted entirely, and the remaining core of four bipolars is replaced with MOS devices. The resulting circuit has a significant amount of nonlinearity, and much effort is made to cope with it. The quarter-square technique is based on the following observation:

$$4\Delta V_1 \Delta V_2 = (\Delta V_1 + \Delta V_2)^2 - (\Delta V_1 - \Delta V_2)^2 \tag{3.75}$$

and hence a multiplier can be realized using two differential squarers, of which there are a plethora of voltage in current out versions.

However, all of these approaches are much more complicated and area-consumptive than we can afford. It was settled upon to use a MOS version of the four-transistor Gilbert core, shown in Figure 3-37, due to its extreme simplicity. Starting as before, we can derive the DC transfer characteristic of the MOS Gilbert multiplier above threshold. We normalize all of the differential variables:

$$\beta_{in} \equiv \frac{\Delta I_{in}}{I}, \;\; \eta \equiv \frac{\Delta V_{in}}{V_r}, \;\; \beta_{out} = \frac{\Delta I_{out}}{I} \tag{3.76}$$

where

$$V_r = \sqrt{\frac{I}{\mu\left(C_{ox} W/2L\right)}} \tag{3.77}$$

$$-1 \leq \beta_{in}, \beta_{out} \leq 1 \tag{3.78}$$

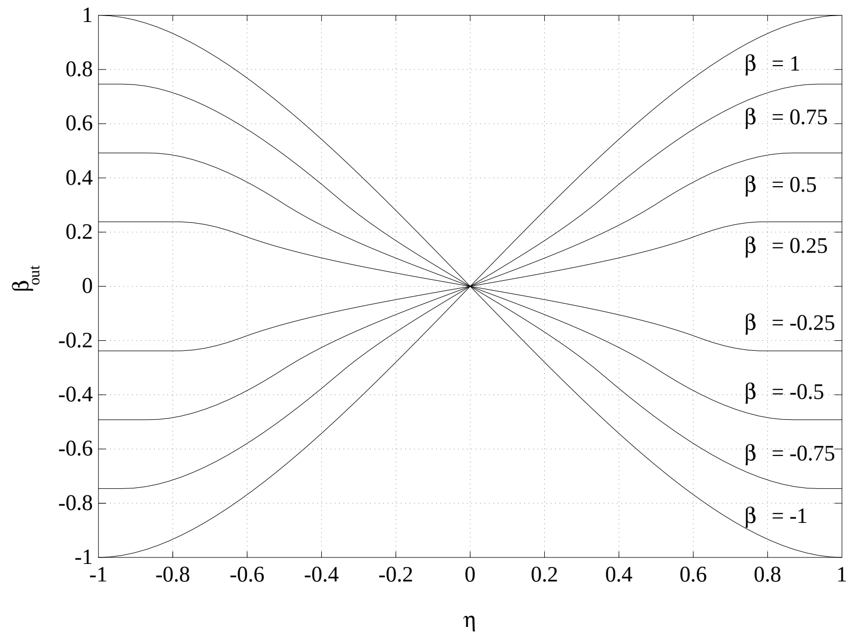


FIGURE 3-38: Normalized input differential voltage to output differential current transfer characteristic of the MOS multiplier.

The normalized output is $\beta_{out} = \beta_1 + \beta_2$ where

$$\beta_1 = \begin{cases} -\left(\frac{1+\beta}{2}\right) & \text{if } \eta \leq -\sqrt{\frac{1+\beta}{2}} \\ \eta\sqrt{1+\beta-\eta^2} & \text{if } \eta^2 \leq \frac{1+\beta}{2} \\ \left(\frac{1+\beta}{2}\right) & \text{if } \eta \geq \sqrt{\frac{1+\beta}{2}} \end{cases} \tag{3.79}$$

$$\beta_2 = \begin{cases} \left(\frac{1-\beta}{2}\right) & \text{if } \eta \leq -\sqrt{\frac{1-\beta}{2}} \\ -\eta\sqrt{1-\beta-\eta^2} & \text{if } \eta^2 \leq \frac{1-\beta}{2} \\ -\left(\frac{1-\beta}{2}\right) & \text{if } \eta \geq \sqrt{\frac{1-\beta}{2}} \end{cases} \tag{3.80}$$

The family of curves defined by these relations is shown in Figure 3-38. While this function has the appropriate qualitative behavior required of a four-quadrant multiplier, it does exhibit substantial nonlinearity. In the case of the source coupled pairs used in our transconductors, the input signals were small (typically $|\Delta V_i| \leq 1V$), so we could have made the gates long enough to accommodate whatever input linearity we desired. The multipliers, though, will potentially need to operate with a wide variety of differential input voltages, depending on the voltage encoding of the position signals. For example, with a 64×64 system and a voltage encoding of $100mV$ per pixel, the required range of the multiplier is $\pm 6.4V$. On the other hand, we can restrict the maximum and minimum voltages of the position to lie in a $1V$ region just as easily. Hence, the input range of the multipliers V_r was arbitrarily set at $1V$. Since the common mode current of the input differential currents is $10\mu A$ at this stage of the processor, this required a $W/L = 1 = 8\mu m/8\mu m$. The full circuit implementation of the first layer of multiplication is shown in Figure 3-39. The resulting output currents from each multiplier is summed and mirrored with a cascoded current mirror into the second stage of multipliers, shown in Figure 3-40.

Since each multiplier potentially provides a maximum differential current of $\pm 2I$ with a common-mode current of $2I$, one could imagine that the output differential-mode would have a maximum of $4I$ with a common-mode of $4I$. In fact, this is not the case because E_x and E_y are orthogonal linear combinations. The output current represents $\Delta p = (x - x_0)E_x + (y - y_0)E_y = \Delta x E_x + \Delta y E_y$. Normalizing and referring back to the δs that form (E_x, E_y), we note that $(\Delta x, \Delta y, \delta_1, \delta_2, \delta_3, \delta_4)$ is confined to a 6D unit hypercube, and we would like to find $\max(\Delta p)$ over this hypercube. From linear programming [67], we know that if a solution exists, it lies at the vertices of the hypercube. Hence, we can evaluate Δp at the vertices to find the maximum. Examining the allowed (E_x, E_y) due to the vertices in δ-space, we note that the only allowed (E_x, E_y) vertices are $(\pm 0.5, \pm 0.5), (\pm 1, 0), (0, \pm 1)$ but $(E_x, E_y) \neq (\pm 1, \pm 1)$. Given the reduced set of (E_x, E_y) vertices, clearly $\max(p) = 1$ and not 2. Hence, while the output common current in the sum is now $4I$, the maximum output differential current is half of that.

In the second layer, we take the output from the first layer and transduce it to voltage. This differential voltage is then applied to the inputs of the multipliers. To accomplish this transduction, the simple I/V converter based on triode connected transistors is used, as shown on the left of the figure. In the triode region, the I/O characteristic of a transistor becomes:

$$I_d = \kappa \left[2(V_{gs} - V_t)V_{ds} - V_{ds}^2 \right] \tag{3.81}$$

and this is true when the gate drive $V_{gs} - V_t > V_{ds}$. Since the gates of the PMOS transistors are grounded, this condition is always satisfied. If we drop the quadratic term in Equation 3.81, we can view a transistor operating in the triode region as a resistor of value:

$$R = \frac{1}{2\kappa(V_{gs} - V_t)} \tag{3.82}$$

Each of the leg currents of the differential input currents $I_+ = 2I + 2\Delta I_p, I_- = 2I - 2\Delta I_p$ is passed through a triode transistor with resistance R and hence

$$\Delta V_p = RI_+ - RI_- = R(2I + 2\Delta I_p) - R(2I - 2\Delta I_p) = 4R\Delta I_p \tag{3.83}$$

And this ΔV_p is input to the second layer of multipliers, which are identical to those of the first layer. Since the differential current is restricted to half of the bias, then $|\Delta I_p| < I/2$ and the maximum voltage swing is $\Delta V_p = 2RI$.

The diode at the top of the triode transistors is sized $W/L = 0.5 = 6\mu m/12\mu m$ to shift the common mode of the output down into the input common-mode range of the multipliers. The common-mode of the input current is $4I = 20\mu A$, and hence the voltage drop across the diode is $\Delta V = V_t + \sqrt{4I/(\kappa W/L)} \approx 3V$. The source voltage is $V_d = V_{dd} - 3V = 7V$ and hence the gate drive is $V_d - V_t \approx 6V$. Matching the maximum output of the I/V converter of $2RI$ where $I = 5\mu A$ to the one volt input range of the multipliers requires $R = 100k\Omega$. With a gate drive of $6V$, this would require a device size of $W/L = 1/12$. However, at the large gate drive we are using, the transistors exhibit a factor of 2 mobility reduction due to mobility degradation [68]. This increases the resistiveness of the devices, so only half the length is required. Hence, the triode transistors were sized at $W/L = 1/6 = 6\mu m/36\mu m$ to provide the appropriate resistance.

The two differential current outputs from the second layer of multipliers are the output error channels e_x and e_y respectively, which are summed up the column in KCL and sent off-chip. Even though the I/O transfer function of the analog processor is multidimensional (there are 10 differential inputs), we can find a transfer characteristic from the transconductor inputs to the output of the second layer of multipliers by for example driving E_y maximally (i.e. $\delta_1 = \delta_2 = \delta_3 = \delta_4 = \zeta/4$). Since E_x and E_t are orthogonal linear combinations to E_y, they are zero under this condition. A mesh plot of the resulting DC transfer curve is shown in Figure 3-41. $(y - y_0)$ is denoted by the normalized variable η while E_y is denoted by the normalized variable ζ. Taking slices in the ζ direction (Figure 3-42) shows the desired quadratic behavior with E_y, while slices in the η direction (Figure 3-43) shows the transconductor behavior with $(y - y_0)$.

3.5.4 The Position Encoder

All of the previous discussion relied on the analog processors having the appropriate voltages encoding the positions (x, y) available at the right time. To accomplish this, a position encoder was designed. The scheme is shown in Figure 3-44. The voltage on a resistive chain is used to encode the y position along the array. A CMOS digital shift register is utilized to select the appropriate x value. Initially, the register has a logic 1 stored in the LSB, while all the rest of the bits are logic 0. This logic 1 is successively shifted up the shift register, 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. Note that using the resistor chain in this fashion for both x and y guarantees that they have the same encoding. When all of the column data has shifted out and the processing is done, the shift register is reset. The design of the digital shift register is virtually identical to the one used in the masking register and was shown in Figure 3-32.

Figure 3-45 is the operational amplifier designed to buffer the position encoder output as shown in Figure 3-44, both for driving off-chip as well as the on-chip multipliers. It is based on a fairly standard MOS two-stage topology [69], where compensation is done using a source follower and a feedback capacitor. 10pF was used as a conservative estimate of the capacitance the operational amplifier will have to drive. Figures 3-46 and 3-47 show the SPICE calculated transfer curves for the design, with a unity gain frequency of 3.5MHz and a phase margin of 50°. The step response in Figure 3-48 indicates a settling time of approximately 500ns, which

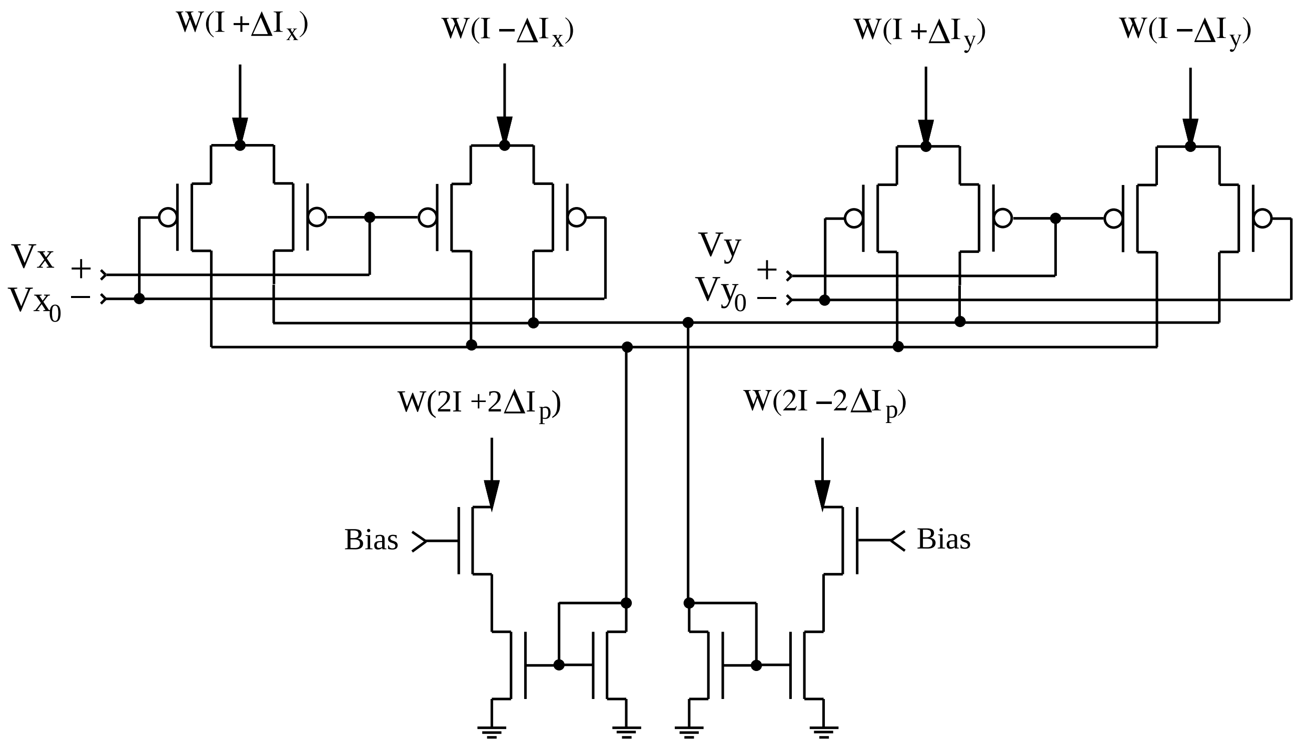


FIGURE 3-39: The first layer of multipliers with output mirroring.

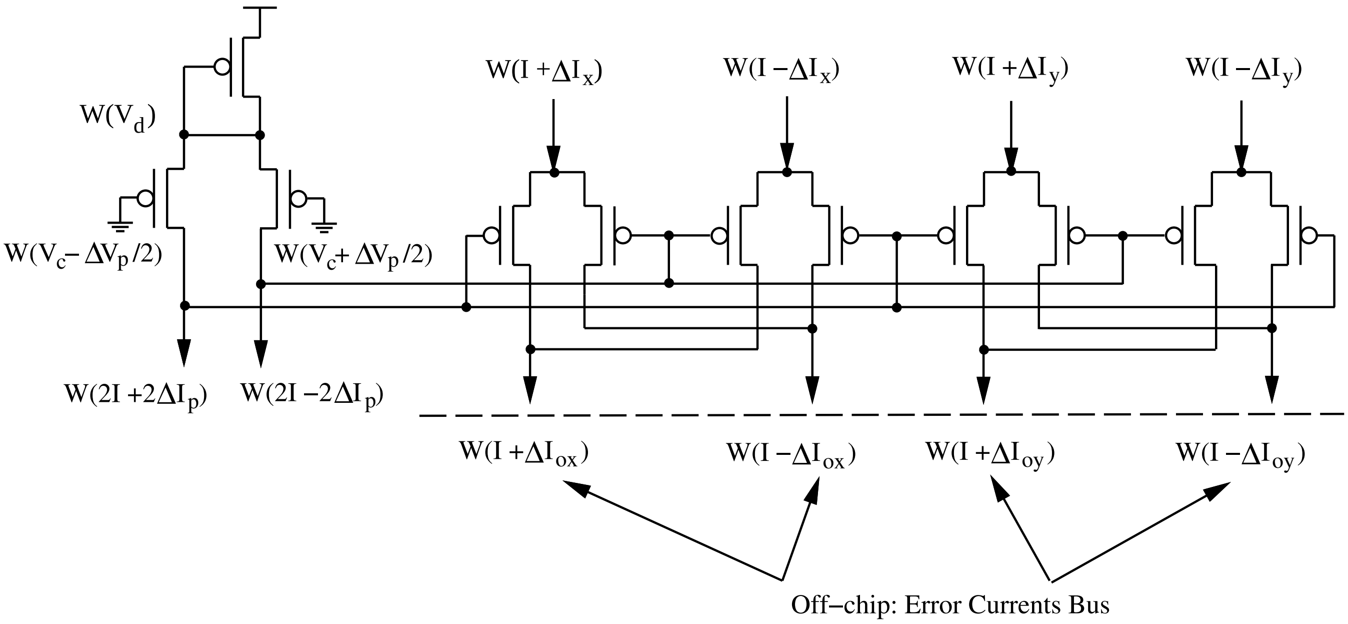


FIGURE 3-40: The second layer of multipliers with triode-based I/V converter.

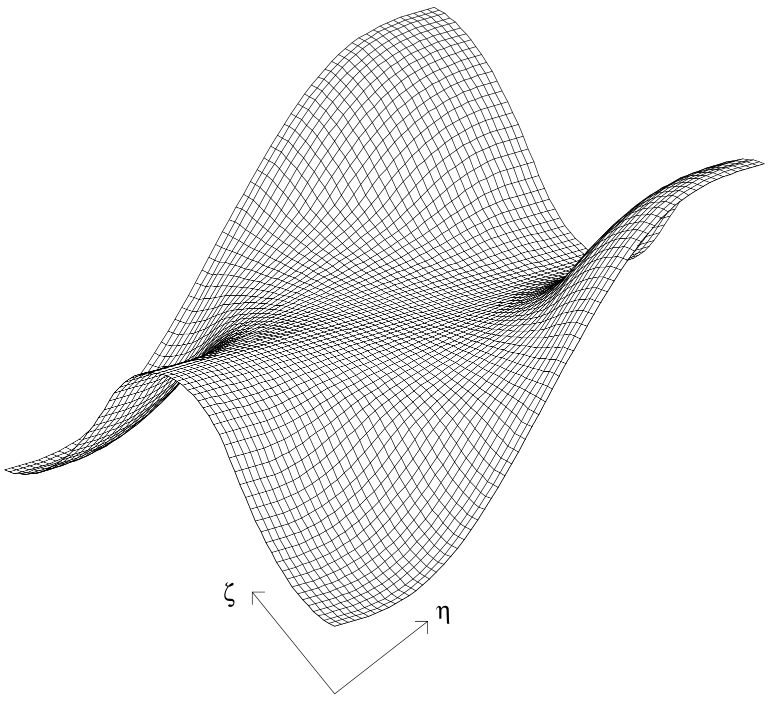


FIGURE 3-41: Mesh plot of the normalized DC transfer curve of the analog processor found by driving E_y (ζ) maximally.

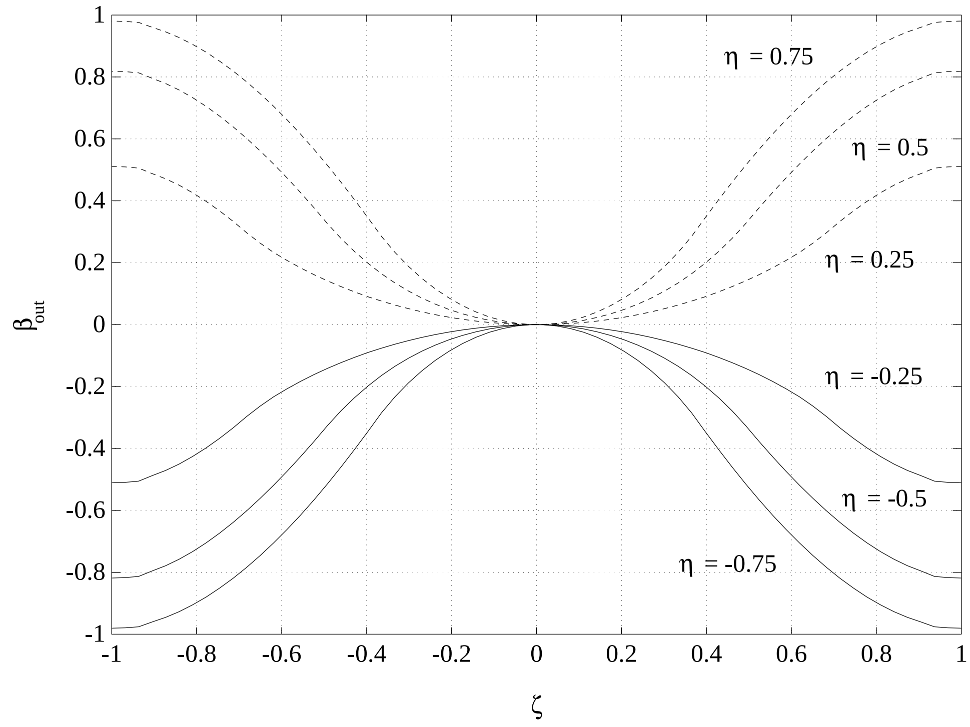


FIGURE 3-42: Family of curves obtained by slicing the surface in the ζ direction.

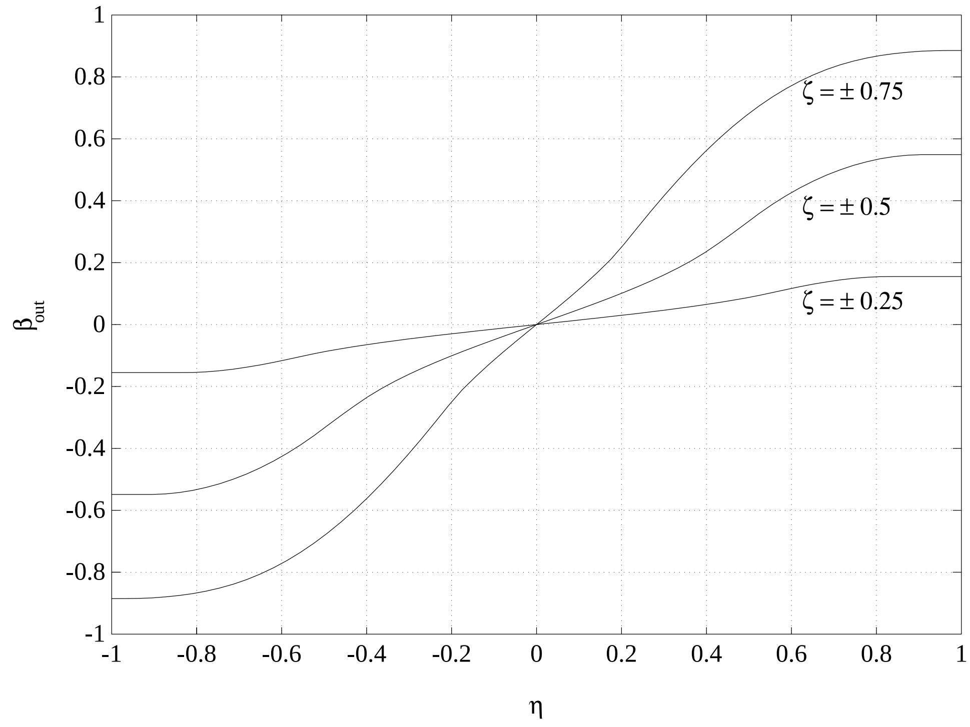


FIGURE 3-43: Family of curves obtained by slicing the surface in the η direction.

is well within our timing requirements.

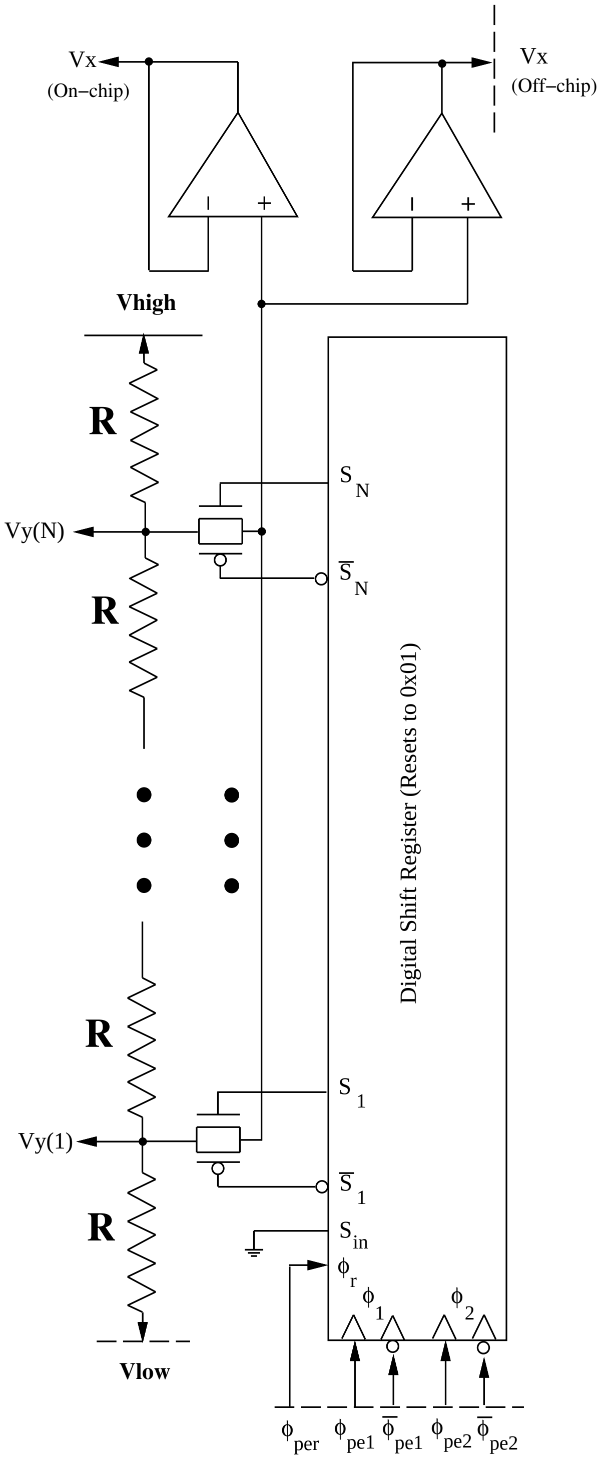


Figure 3-44: Circuit diagram of the position encoder.

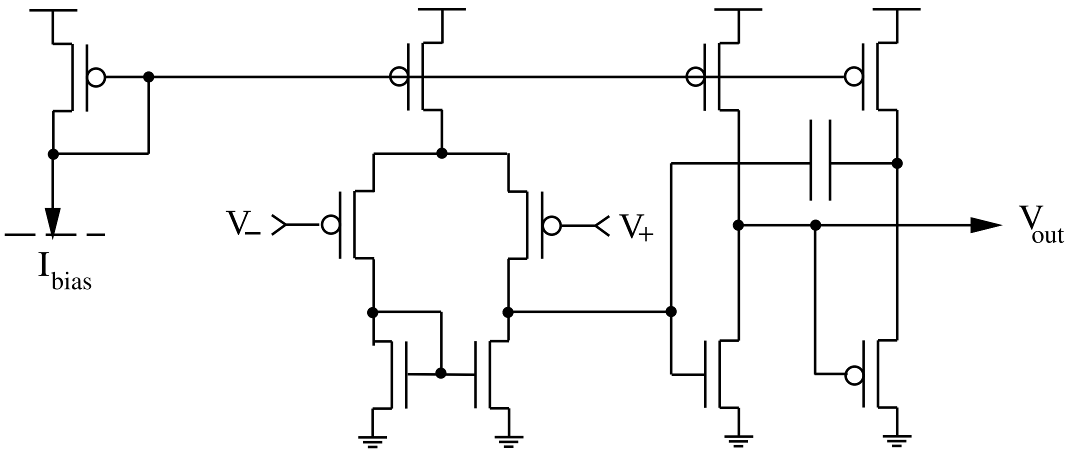


FIGURE 3-45: The simple two-stage operational amplifier used in the position encoder.

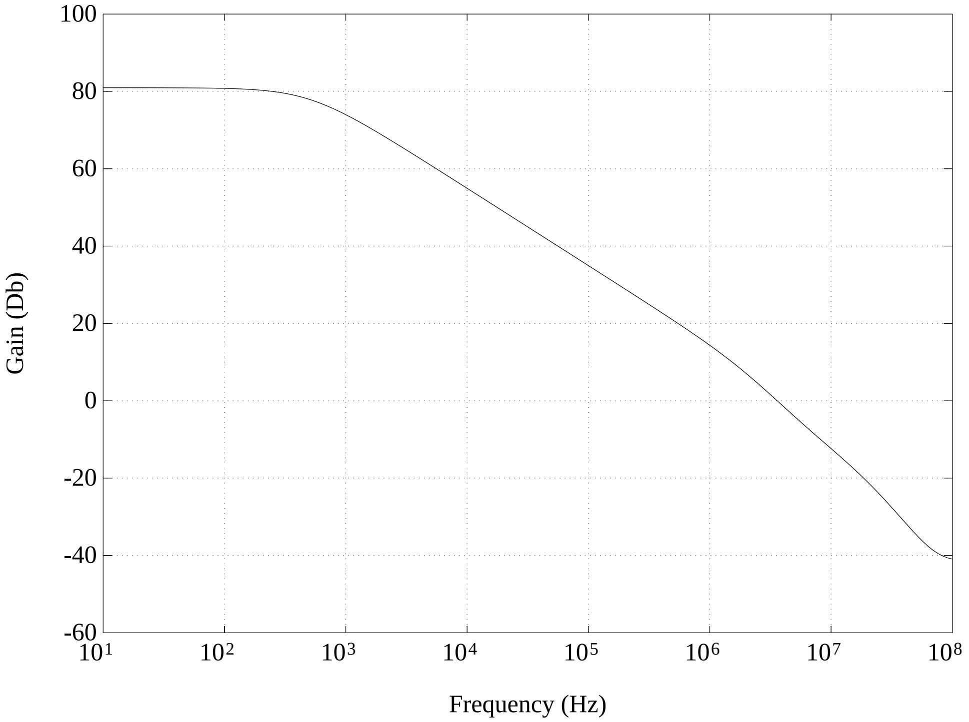


FIGURE 3-46: SPICE gain response of operational amplifier used in multiplexed position encoder. C_L = 10pF.

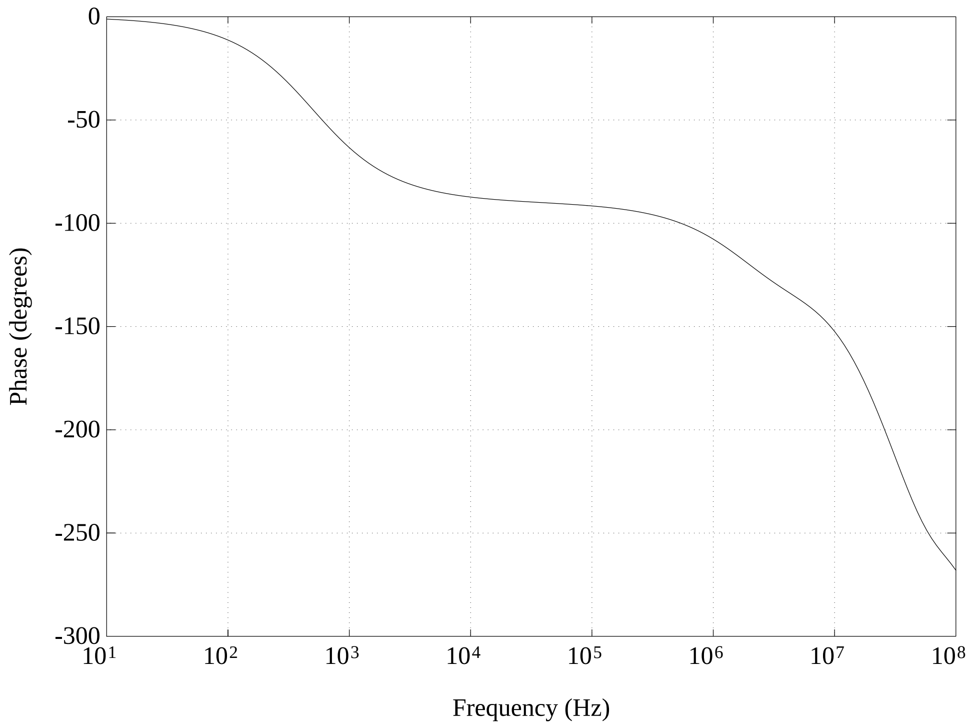


Figure 3-47: SPICE phase response of operational amplifier used in multiplexed position encoder. $C_L = 10$pF.

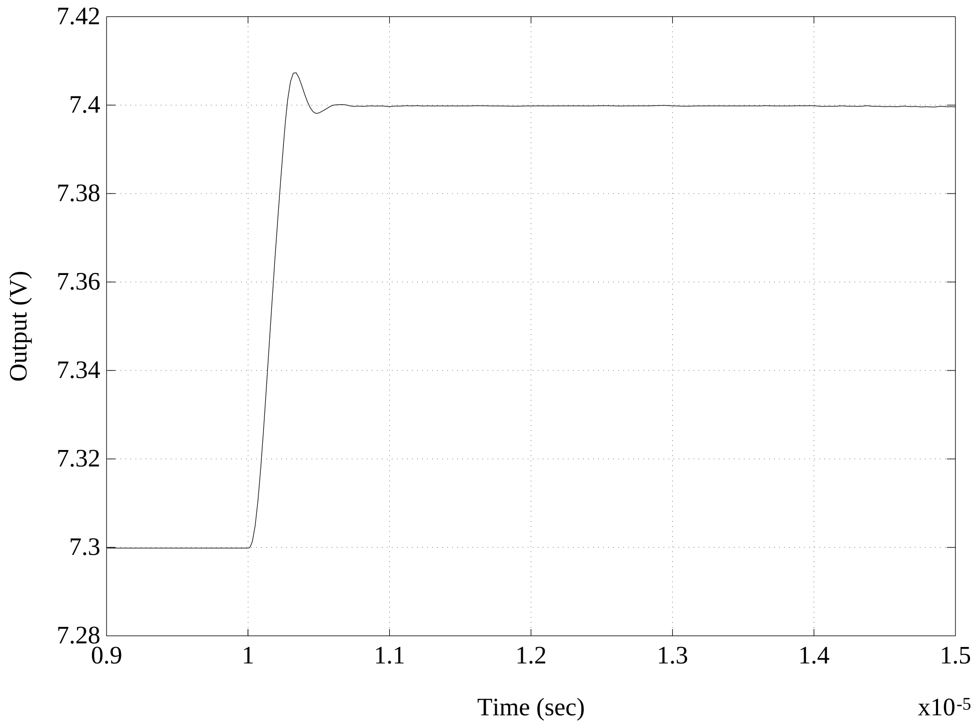


Figure 3-48: SPICE step response of operational amplifier used in multiplexed position encoder. $C_L = 10$pF.

4

The FOE Test System

4.1 The Motion Problem

We now turn to the problem of testing the FOE chip. The ultimate aim of this thesis is to demonstrate the FOE chip operating in real time with real motion. Testing the FOE chip with real motion is quite a challenging physical goal. The premise of our algorithmic approach is that the camera, in our case the chip, is undergoing translational motion. This is so that all the brightness variations seen by the chip are only the result of its own motion. The test board to support the operation of the FOE chip is quite complex in order to provide sufficient testing flexibility. Of course this would normally not be the case in a final customized system, but in the prototyping stage having sufficient on-board resources to explore all aspects of the chip's operation is essential. Hence, the board is large and cumbersome; moving it is not a feasible approach for the experimental setup. Separating the chip from the board in order to move the chip while keeping the board motionless raises a whole host of interfacing problems between the chip and the board and hence it was decided that this approach was also to be avoided. The last option considered was to optically produce images corresponding to ego-motion and introduce them to the chip without actually moving the chip.

There are several ways that we considered for generating images corresponding to motion and inputting them optically into the FOE chip. Showing the chip a movie, either motion picture or video, is not very feasible; we would have to then synchronize the system to the frame rate of the movie. Another idea was to use some clever trick with mirrors instead; for example moving a mirror in front of the chip generates apparent ego-motion. But this also becomes difficult due to the size of mirror required for reasonable fields of view. The solution that was decided upon for testing was to use a flexible fiber-optic image carrier. Image carriers made of flexible bundles of optical fibers can be used for the passive transmission of images. In our test setup, we move the tip, along with its lens system, of such a carrier with a known, calibrated motion while the near end is held fixed, focusing the resulting images onto the motionless chip in the

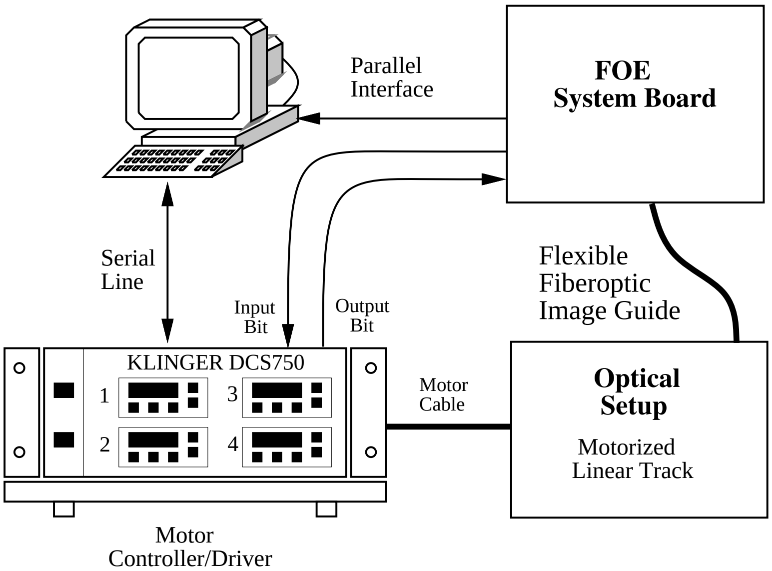


FIGURE 4-1: A high level view of the basic test setup, showing the two major subsystems: optical/mechanical, and electrical.

test board. This provides the chip optically with images corresponding to ego-motion without the chip actually having to move.

A basic high level view of the testing setup is shown in Figure 4-1. The test system is broken up into two distinct parts. The optical/mechanical subsystem consists of the image carrier with its optical interface to the chip, along with the motor system for moving the tip of the cable with calibrated motion. The electrical subsystem consists of the circuit board for fully testing the FOE chip. A host computer oversees the overall functioning of the entire setup; communication with the test board is accomplished through a parallel interface, while communication with the motor controller is done with a serial port. Finally, through the use of the general purpose input and output bits available on the motor controller, synchronization between the resulting motion and the image acquisition of FOE chip can be maintained.

4.2 The Optical/Mechanical Subsystem

4.2.1 The Flexible Image Carrier

An optical fiber is constructed of a *core*, typically made of GeO_2 and SiO_2, with refractive index n_1, and an outer layer, *cladding*, made of SiO_2 with refractive index n_2. Light falling onto a fiber within the angle φ is transmitted down the fiber by total internal reflection as shown in Figure 4-2a. This angle is given by the *numerical aperture*:

$$\sin(\varphi) = \sqrt{n_1^2 - n_2^2} \tag{4.1}$$

Image carriers are basically bundled arrays of optic fibers, as shown in Figure 4-2b. The ends

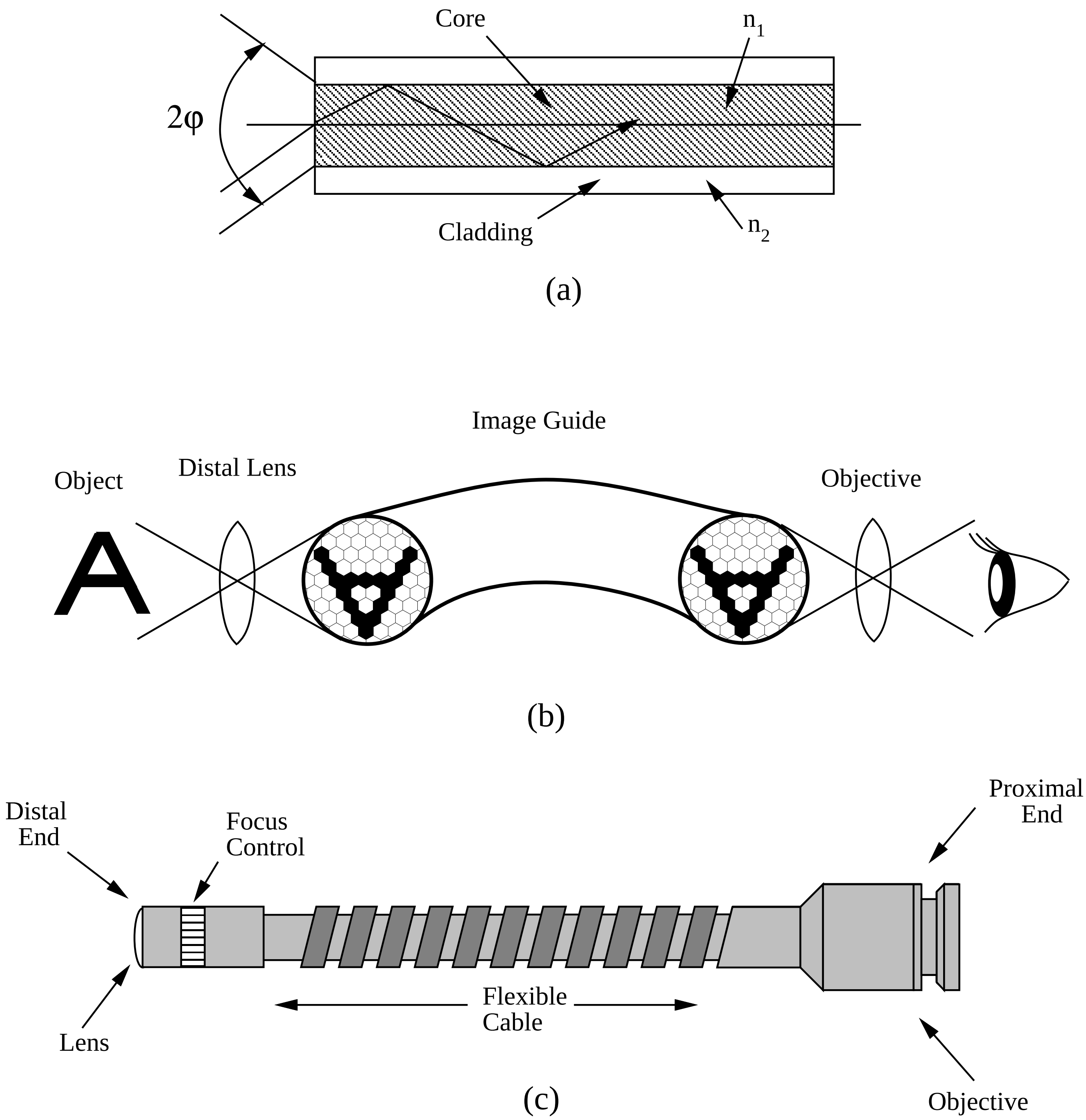


FIGURE 4-2: a). Light within the critical angle φ given can be transmitted down an optic fiber using total internal reflection. b). The principle of image transmission by an image guide. c.) Flexible fiber-optic carrier used in remote visual inspection.

Fiber Diameter:	$9 - 11\mu$m
Bundle Size:	30000 – 40000 fibers
Cable Outer Diameter:	4 – 20mm
Loss:	≤ 0.2dB/m in the visible
Field of View:	$20° - 80°$
Depth of Field:	10mm$-\infty$
Cable Bend Radius:	1 – 4 inches
Cable Length:	0.5 – 2.5m

Table 4-1: Typical image carrier parameters.

of the fiber bundle are fused together and then cleaved, forming image planes. With the array ordering the same at both ends, an image focused by a lens system on one end of the carrier is transmitted passively down the fiber bundle and out the other end. Inside the carrier, these fibers are flexible, and so the entire carrier between the two ends is also flexible. The standard carrier shown in Figure 4-2c has a far, or *distal*, end with a lens system, which may be of fixed or variable focus. The fiber bundle in the cable is protectively sheathed, typically with steel coils, steel mesh, and waterproof plastic tubing to protect it from harmful environments as well as to prevent damage from forces due to excessive bending or crushing. The near, or *proximal*, end has an objective through which the images can be viewed, much as in a monocular microscope.

These carriers are normally used for remote visual inspection of places that are not routinely accessible, for example the inside of jet turbines. Since the areas may be low in ambient light, image carriers are typically equipped with light pipes so that illumination can be introduced into the area being examined. Additional enhancements such as varying degrees of articulation to move the distal tip and remote focusing control of the distal lens system from the proximal end are available. In our experimental setup, we use a stripped down version without these additions.

Table 4-1 shows typical parameters for commercially available image carriers [70]. The individual fibers in the carrier have a diameter of about 10μm and typical bundle sizes are in the 30 thousand fiber range. When interfacing to a commercial CCD chip, which typically has > 500 thousand pixels, aliasing can be a substantial problem, as the pixelation of the cable will beat with the pixelation of the imager. Additionally, the cladding between individual fibers can be visible, leading to fixed pattern noise. However, the FOE chip has only 4 thousand pixels, and hence this is not a considerable problem, as these cables have an order of magnitude more fibers than the chip has pixels. Placement of an aperture stop in the optics can be used to low pass filter this noise away if necessary. Fabrication of the bundle is done by drawing the fibers out under high temperature and pressure. As a result, the bundle packing is hexagonal, as was shown schematically in Figure 4-2, and the overall bundle is circular. Hexagonal is optimal for

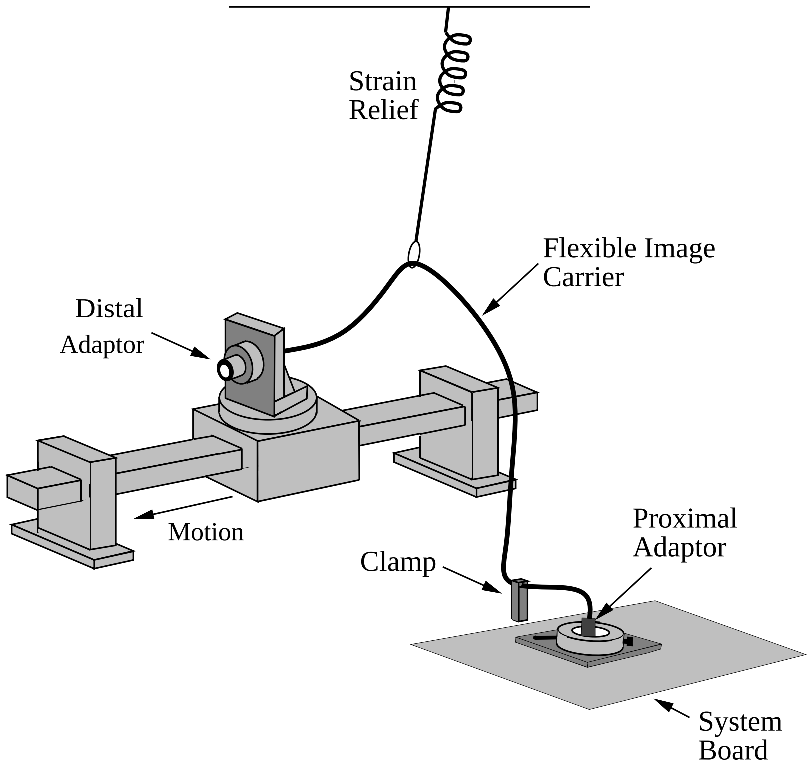


Figure 4-3: The physical structure of the mechanical and optical system.

close packing and hence gives maximum fill factor, which can be as high as 98% for a good cable. The outer diameter of the cable is substantially larger than the bundle itself, due to the protective sheathing, but cables as small as 4mm in diameter are available for insertion through extremely small openings.

Loss for a silica optic fiber is typically quoted as being less than 0.2dB/m in the visible spectrum. In practical terms, this means that over and above the 40-50% loss that the transmission through the glass in the fibers entails, an additional loss of 10% can be expected from a 2 meter long cable. Overall, a loss of perhaps a factor of 2 is to be expected. The algorithm for estimating the FOE performs better with wide angle lenses, and with suitable optics a fairly wide field of view with a half angle $\theta_v = 40°$ is available. With a bend radius of several inches and lengths ranging up to 2 meters, the image carrier appears ideal for use in motion experiments in the lab.

4.2.2 The DC Motor System

A physical representation showing more explicitly the structure of the mechanical setup of the test system is shown in Figure 4-3. We use a Klinger Scientific DC motor system based on motion down a linear track. The GV88 long travel rail is elevated by supports on either end which are in turn rigidly mounted to a Newport optical table. On this track is a carriage which

Repeatability:	4μm
Absolute accuracy:	100μm over $1m$ travel
Useable travel:	1 meter
Maximum Velocity:	0.5 m/s
Maximum Acceleration:	0.8 m/s^2 unloaded
Controller Features:	Trapezoidal Velocity Generator Real-time Motion Profiling External Event Synchronization Independent Program Execution
Controller Conversion:	10μm/count

Table 4-2: Specifications of the Klinger motor system used in the experimental setup for testing the FOE chip.

moves along the rail using a rack and pinion system driven by a UE72CC DC motor mounted underneath. The motor is controlled by a Klinger DCS750 motor controller using positional feedback from a shaft encoder. The controller implements a discrete time PID feedback control system using National Semiconductor's LM628 servo controller chip. The overall specifications of the systems are shown in Figure 4-2.

The image carrier used in the FOE system is the Olympus ICA36A-20. It has a working length of 2 meters, a bundle size of 40 kfibers, a $10mm$ outer diameter, a field of view of $2\theta_v = 80°$ and a focus control at the far end. The distal end of the flexible image carrier is rigidly affixed to the moving carriage, while the proximal end is held stationary over the chip by the proximal mount. In order to keep the cable out of the way and prevent damage during motion, strain relief in the form of bun-gee cord is provided. As the carrier is drawn back, the cord pulls the cable away from the motor system. To eliminate the transmission of vibration to the optical interface at the chip during motion, a clamp based on a Melles Griot V-block holder is placed on the cable just before it reaches the board. The proximal interface, which we will discuss shortly, is not completely rigidly attached, and the varying tension on the cable during motion moves the optics slightly and hence distorts the received images at the chip. The V-block holder is used to grip the cable without exerting undo force on it and prevent this distortion.

4.2.3 The Distal Interface

Clearly, the design of the distal and proximal interfaces for the cable are critical. The distal end must rigidly grasp the cable tip, allow access to the focus control, as well as provide the means to place the direction of viewing relative to the motion direction and hence allows control of the placement of the FOE in the image plane. Figure 4-4 shows the constructed distal mount in detail. Affixed atop the carrier is a Klinger TR120 large rotation stage. This stage has 360°

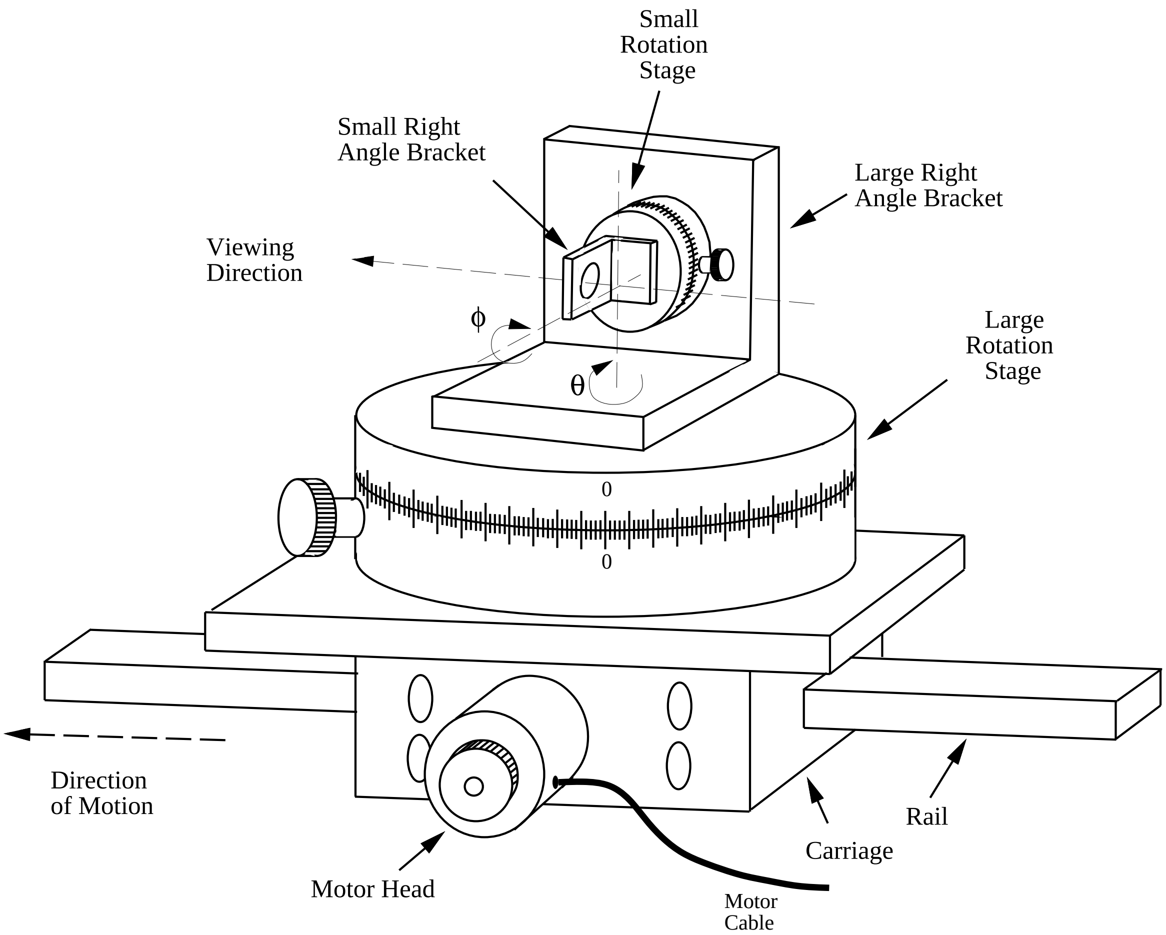


Figure 4-4: The distal mount, allowing placement of the FOE anywhere in the image plane.

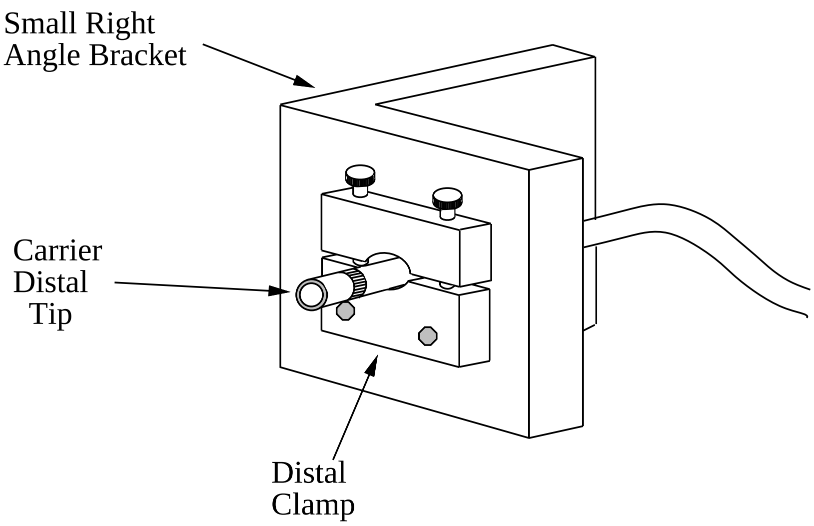


Figure 4-5: The distal clamp for attaching carrier to the distal mount.

of rotation and corresponds to a rotation θ about the vertical axis. Setting of this angle can be done to within an arc-minute, and a set screw is provided to maintain the setting during motion. On top of the TR120 is a large right angle bracket, and affixed to the inner face of this bracket is a TR80 rotation stage, which is a smaller version of the TR120 complete with all the same features. The position ϕ of this small stage corresponds to a rotation about the horizontal axis *of the large bracket*. It should be noted that changing θ changes the direction of this axis of rotation. A small right angle bracket is finally placed on the face of the TR80. Through the settings of θ and ϕ, the viewing direction defined by the opening in the small right angle bracket can be arbitrarily set; this allows the positioning of the viewing direction relative to the motion along the track. Note that in the figure the rotation axes are shown as intersecting. The brackets were constructed to achieve this as pure rotation of the camera is necessary to use the calibration by rotation method which we will discuss in Chapter 5. Camera calibration is necessary to map the motion direction we have set up to the actual FOE location on the image plane.

Attachment of the tip of the cable to the distal mount is shown in Figure 4-5. The distal end of the cable with its lens and focus control extends outward from an opening between the two plates mounted on the small right angle bracket. The plates are screwed together, pinching the cable at the metal flange just past the focus knob. This rigidly holds the cable in line with the center of the bracket.

4.2.4 The Proximal Interface

At the proximal end, we need a method of getting the images to the FOE chip. Olympus provides adaptors to connect their image carriers to video cameras, and this is where our design begins. We use the MC-08 adaptor which is intended for use with a 2/3 inch CCD imager. As shown in Figure 4-6, this adaptor fits over the objective and provides a C-mount end suitable for attachment to a video camera. The "2/3 inch" is a misnomer; it is a holdout from the old Vidicon tube days and is a measurement of the outside diameter of the imaging tube. The 2/3 inch format corresponds to an active imaging area of 8.8mm$\times$6.6mm. This is sufficiently similar to the 6.9 mm$\times$6.9mm of active imaging area in the FOE chip that this adaptor is close in magnification to what we need. If one actually attaches the adaptor to a video camera, the image provided by the carrier does not fill up the entire screen. The fiber bundle is circular, due to its method of fabrication, and the imaging chip in the video camera is rectangular. Hence, the magnification in the adaptor is set so that the boundary of the circular image fits entirely on the CCD imaging chip with a significant amount to spare. The magnification is most likely kept even lower than necessary to avoid aliasing problems with the imaging chip. For the FOE chip, it is obviously important that no part of this boundary be visible to the imager, and hence it was estimated that an additional magnification was required. With the correct magnification, the circular spot from the carrier would fill the imager entirely, and as a consequence $1-2/\pi = 36\%$

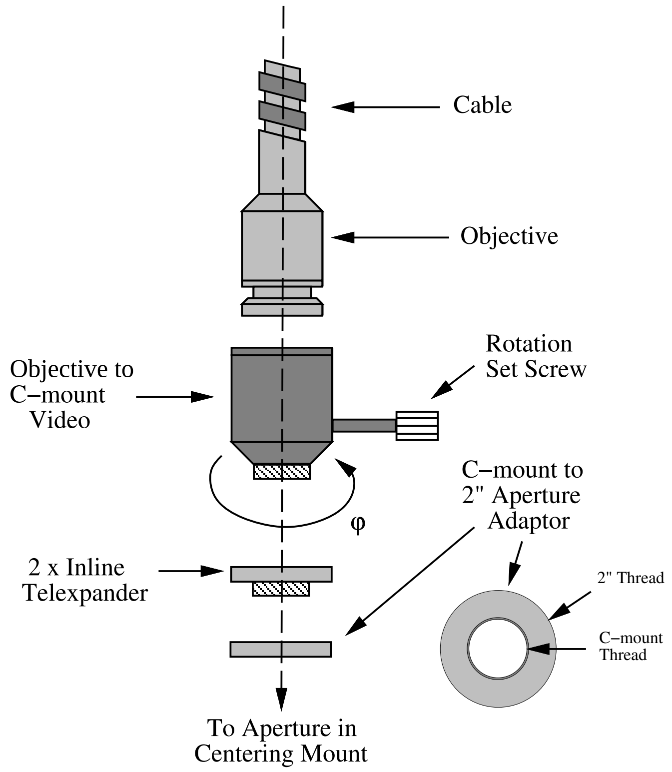


Figure 4-6: Optical interface to the proximal mount.

of the data from the carrier would be lost, solely due to fitting a square inside a circle. The required magnification is between 1 and 2. Anything between 1 and 2 would be a nonstandard magnification, and without resorting to custom optics a conservative choice is a 2× expander. An in-line version of a C-mount 2× telexpander from Cosmicar was placed at the output of the MC-08.

This magnification is based on a focusing distance between the lens and the image plane given by the C-mount standard of 0.69 inches. Hence, we need to place the proximal optic assembly over the chip at this distance. Additionally, the design of this connection should be such that we are able to position the optic axis arbitrarily in the image plane as well as provide focusing control to achieve the correct distance over the chip. The final solution is shown in Figure 4-7. The proximal adaptor combines an Oriel 17330 lens-centering mount shown in Figure 4-7a with the adaptor plate of Figure 4-7b. The FOE chip sits in a ZIF socket on the test board; the plate goes around the FOE socket as shown. The lens centering mount is bolted to the plate which is in turn bolted through the board itself, through some spacers elevating the board away from the optic table, and into the surface of the table. This rigidly connects all of the elements involved to the table as demonstrated in Figure 4-7c. The lens-centering mount has a 2 inch threaded opening; the last component shown in the assembly of Figure 4-6 connects the C-mount end of the telexpander to the proximal mount. This final adaptor is a Delrin ring

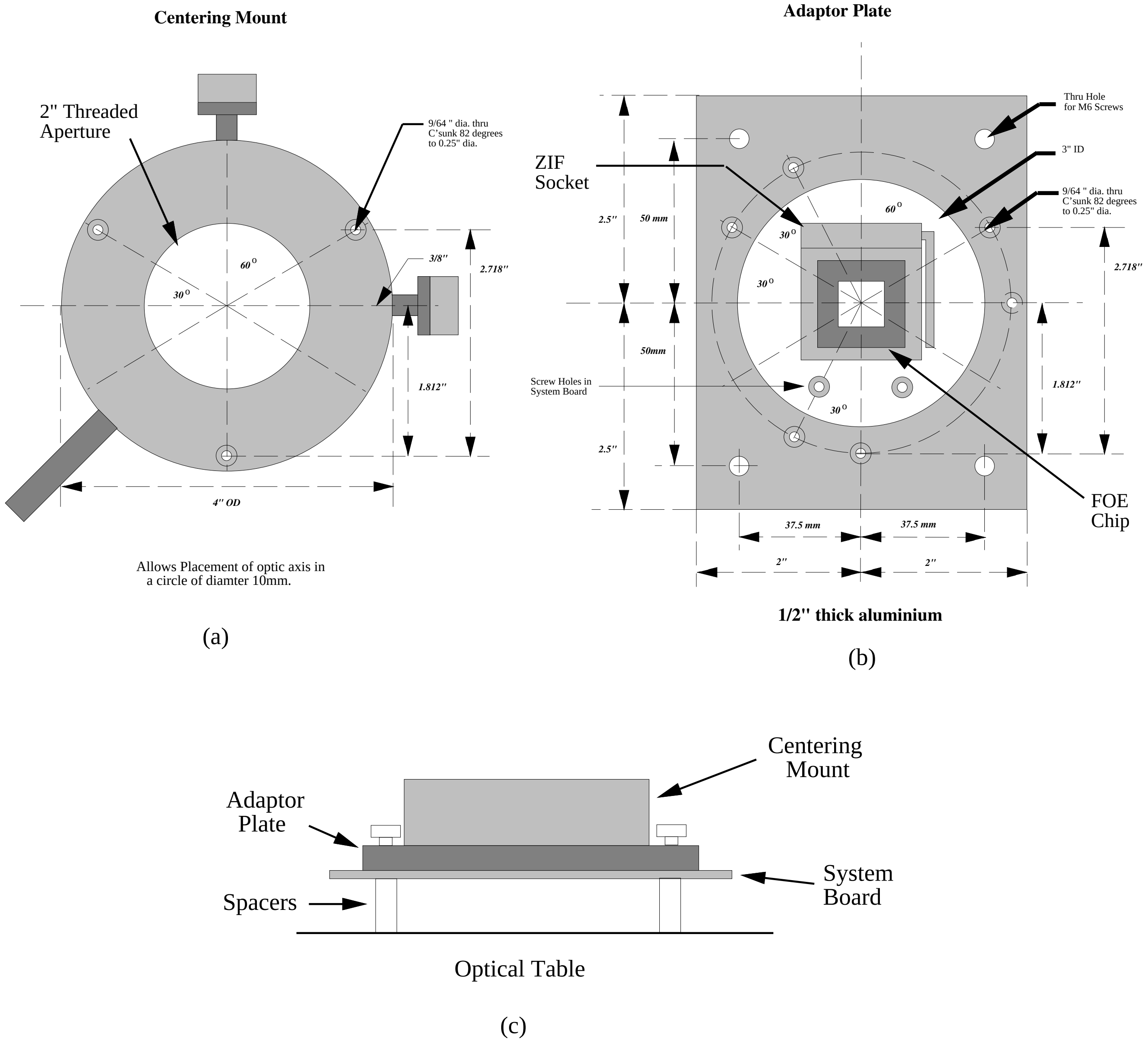


FIGURE 4-7: The proximal end, showing attachment to the FOE board.

with C-mount thread on the inner surface and thread appropriate for screwing into the 2 inch aperture of the lens-centering mount on the outer surface. Taking into account the thickness of the adaptor plate, the ZIF socket, the chip package dimensions itself, etc, the distance from the end of the telexpander is within a few millimeters of being correct; final focusing of the interface is done by screwing the assembly in and out of the lens-centering mount.

Note that the resulting proximal interface actually has 4 degrees of freedom: x, y positioning from the centering mount, z position from screwing the whole assembly up and down in the centering mount aperture, and finally a rotation in the MC-08. This rotation φ is set by a screw in the collar of the adaptor. The non-rigidity of the distal mount is due partly to this rotation knob, which does not set very strongly, and play in the connection to the lens-centering mount. These nonidealities result in the need for the vibration suppression clamp to keep disturbances to the interface at a minimum.

This completes the optical/mechanical design of the FOE test system. Through it, we provide the chip with images corresponding to ego-motion, as well as position the FOE anywhere in the image plane through placement of the viewing direction relative to the motion direction. Of course, in addition to the angles θ and ϕ, the exact position of the FOE resulting from the motion depends not only on the location of the intersection of the optic axis with the image plane, but also on the unknown rotation φ and the principal distance f. These parameters will be determined through camera calibration. Finally, a method for focusing the system is required. This can either be accomplished using raw image data, or perhaps by maximization of the signal s_{quad}, which measures the sum of the squared magnitude of the image gradient.

4.3 The Electrical Subsystem

The electrical subsystem encompasses the printed circuit board fabricated to support the operation of the FOE chip in all its various modes. A combination floorplan and block diagram for the system board is shown in Figure 4-8. There are four main sections to the board: the bias section, the CCD input/output section, the pattern generator, and the feedback control processor (FCP). The FOE chip itself sits in the proximal mount as described in Section 4.2.4, occupying a large central section of board area due to the physical size of the mount.

The bias section to the far right on the board generates the DC currents and voltages required to operate the chip. Additionally, since the FOE chip also has an extra copy of the analog processor used in the main processing array with all of its inputs and outputs broken out separately, the bias section has support to drive these test inputs as well as to measure the resulting test outputs.

Above and below the chip are the input and output drivers for getting the eight channels of CCD data into and out of the imager on the FOE chip. These channels were designed to interface with the Demonstration System for Early Vision processors (DSEV) designed for the

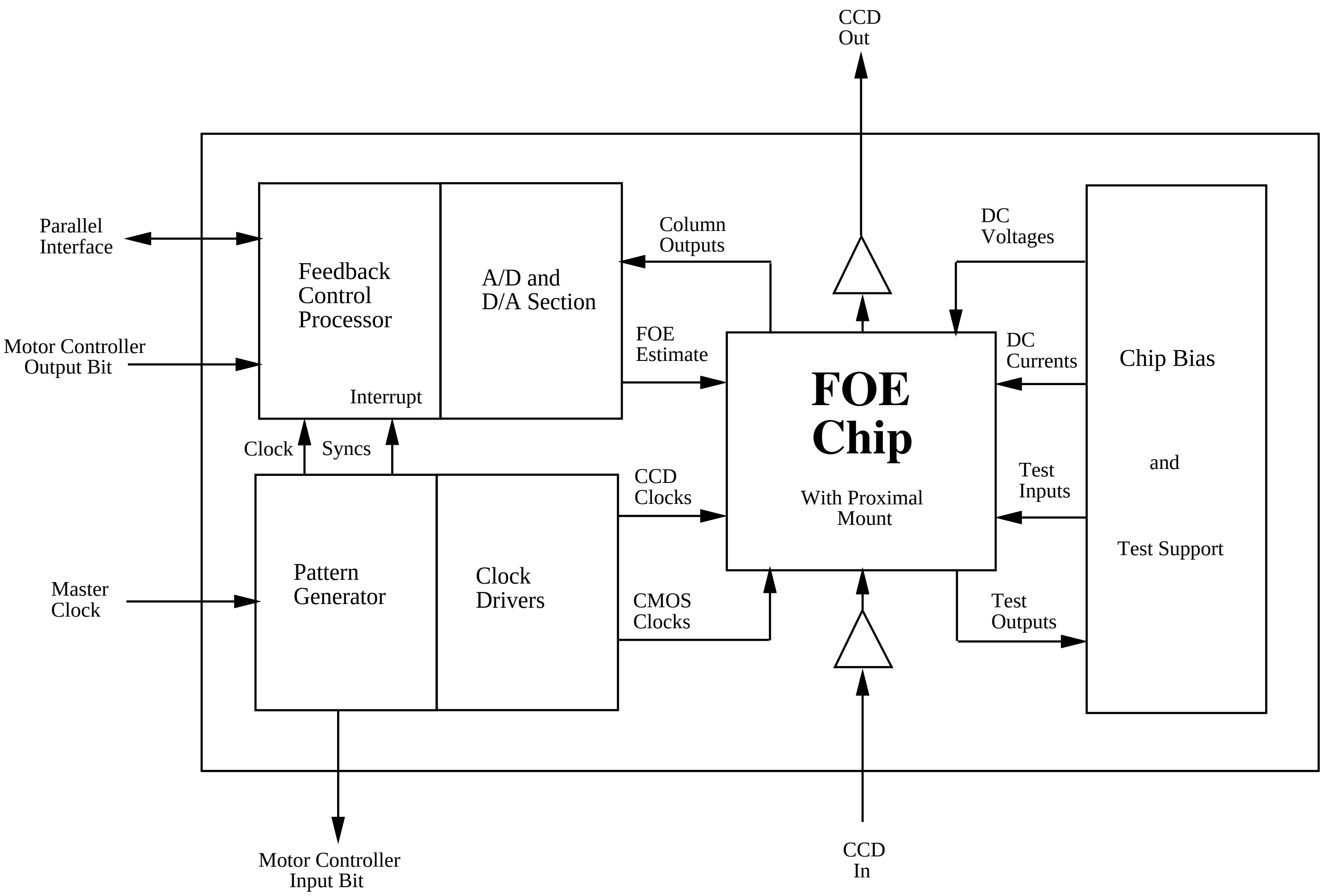


FIGURE 4-8: Block diagram and floorplan of the FOE system board.

MIT Vision Chip Group [71]. The A/D and D/A pods in this system were expected to do the level shifting and scaling necessary for interfacing to test boards built around the high speed analog machine vision chips designed by our group, and hence the on-board input and output drivers on the FOE system board were designed to only buffer these signals. Unfortunately, although the system described in [71] can accommodate 8 channels of throughput, only the 2 channel version was constructed. Hence, when the need arose for raw image data acquisition from the FOE chip, an enhancement daughter board was constructed to accomplish this task and will be discussed in Section 4.3.4.

In the lower left hand corner of the FOE board is the pattern generator which serves as the master timing source for the entire board and creates the necessary clock waveforms to run the FOE chip. These signals are then driven on-chip using clock drivers. In the case of non-critical CMOS signals, these are merely open-collector pull-up drivers, whereas specialized drivers are required for the CCD signals.

The feedback control processor (FCP), shown in the upper left hand corner of the board, consists of the ADC and DAC interfaces to the FOE chip, the DSP hardware for implementing the discrete time feedback loop, and the parallel interface for communicating with the host computer. The FCP is entirely interrupt driven from the pattern generator for correct sequencing of data acquisition, as well as from the motor controller for synchronization of the feedback loop to the motion of the stage, and from the parallel interface for communication and offloading of data.

The latter two sections, the pattern generator with clock drivers and the feedback control processor along with its chip interfaces, form the majority of the board and we will examine each in turn.

4.3.1 Clocking

Generation of clocks is an exceptionally important requirement for all CCD chips. Frequently there are a large number of clocks, often with exceptionally complex clocking patterns. One solution to clock generation is to use a Digital Acquisition System (DAS) or some similar piece of equipment. The drawback to such an approach is usually a limited number of channels, or insufficient algorithmic capacity for generating the lengthy sequences. However, the moderate speed requirement ($\approx$ 10MHz) for clock generation led to a different approach wherein clock generation is done in a reasonably flexible manner entirely on-board. Figure 4-9 shows a schematic representation of the pattern generator designed for the FOE system. A clock source provided by an external pulse generator is driven on-board; this signal forms the master clock 4DC. The master clock is divided by two resulting in the 2DC signal which drives the FCP. 2DC is further divided by two forming the pattern generator dot clock DC.

At the heart of the pattern generator is the IDT49C10A, a 16-bit microprogram address sequencer intended for controlling the sequence of the execution of microinstructions stored in a

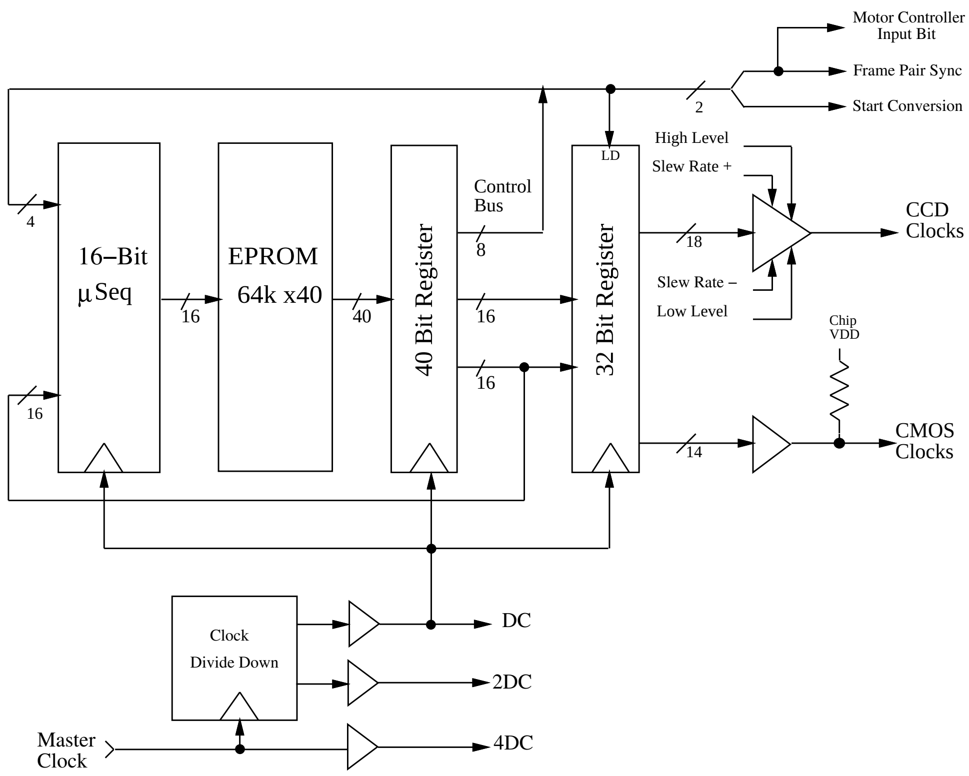


FIGURE 4-9: Block diagram of the pattern generator for generating all clock waveforms on the FOE board.

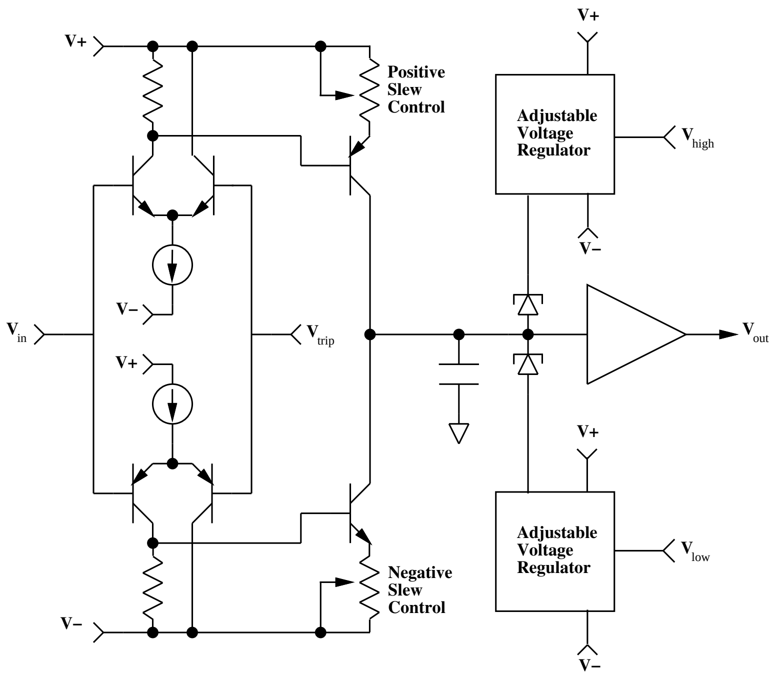


FIGURE 4-10: Sample design for a clock driver with adjustable levels and rise/fall times.

microprogram memory. It incorporates a deep stack providing PUSH and POP instructions as well as subroutine calls and a register/decrementer with a zero detector for conditional looping and branching. The basic operation of the micro-sequencer is to cycle through a 16-bit wide address. As it does so, it accesses the program store in the EPROMs and the resulting 40 bit wide word is stored in a micro-pipeline register. A path back to the micro-sequencer from this register is necessary to provide the addresses required for jumping as well as the 4-bit control nibble for the next instruction. After the micro-pipeline register is the final output register leading to the clock drivers which send the signals to the FOE chip. When an address or a loop count is present at the micro-pipeline register, the output register is frozen for the one cycle necessary to execute the required algorithmic instruction, and hence none of the output signals are affected.

Because of the algorithmic nature of the generator, the length of any particular program was typically quite small. Hence, the 16-bit address space was broken up into 8 banks, selectable using DIP switches on the board. This allowed for quick switching between various clocking modes, and rapid prototyping of clock sequences. The exceptional algorithmic capabilities of IDT49C10A allow the formation of very complex clocking sequences. The board implementation results in 32 bit wide patterns at speeds of up to 12 Mpatterns/sec. The largest limiting factor on this operational speed is the 50ns access time of the Cypress CY7C287 EPROMs that were used. The pattern generator had more than enough power for the generation of all of the many varied clocking schemes needed during the testing of the FOE chip. In order to make the complete design of the clocking sequences more manageable, an assembler and simulator were specifically written for the pattern generator; this greatly eased the testing and debugging process.

Once the desired pattern has been created, the actual clock waveforms are applied to the chip through clock drivers. The digital sections of the FOE board have 5V levels, whilst the FOE chip uses a 10V supply. Hence, the CMOS clocks need to be level shifted up to the supply of the chip. This is accomplished using simple TTL 7406 and 7407 open collector drivers. The desired shapes of the CCD clock waveforms are substantially more complicated, and require the design of specialized clock drivers. Variable levels, both high and low, are needed as well as controllable rise and fall times to enhance transfer efficiency. A sample clock driver design for the FOE board is shown in Figure 4-10. This driver basically functions by appropriately shunting fixed currents on and off an integrating capacitor. This controls the slew rate of the output and a diode clamp is used to fix the output levels. This is quite a large and complex circuit, especially considering that the FOE system requires 20 of them. Thankfully, the Elantec EL2021C Monolithic Pin driver integrates the functionality of this clock driver using a very similar design into a single 18 pin package, with the additional enhancement that the slewing currents are formed using voltage to current converters.

4.3.2 The Feedback Control Processor

The Feedback Control Processor manages the acquisition of data from the FOE chip through its analog to digital converter (ADC) and digital to analog converter (DAC) interfaces, implements the discrete time feedback loop closed around the chip, and transfers data to the host computer for analysis. The system is built around the TMS320E14, a 16-bit microcontroller/processor from the first generation of DSP processors from Texas Instruments. This chip was chosen because it combines the features of a basic DSP processor with a controller in a single architecture. From a DSP standpoint, the chip's 32-bit ALU/accumulator, 16×16 bit multiplier, 0-16 bit barrel shifter, and 256 words of on-chip RAM along with a 160ns execution time provide sufficient performance for many basic signal processing needs, and the simple nature of the computation of the FOE feedback loop would not task these resources. From a controller standpoint, the event manager with capture inputs and compare outputs, a 16-pin bit-selectable output port, a serial port, and four independent timers are highly useful for real-time control of external devices, such as the ADCs, DACs, and asynchronous parallel port on the FOE board. This combination of processing power and control was perfect for use in the FOE board. Unfortunately, the 'E14 was obsoleted during the testing phase of the FOE project as the processor is an old one; TI is currently in its fifth generation of DSP processors, all of which have vastly more processing capability than the 'E14.

The FCP microprocessor system was designed to include the following features:

- An external program store once again using 50ns Cypress CY7C287 64K×8 EPROMs. The addressable space of the 'E14 is 12-bits, and hence the 16-bit address space of the EPROMs was broken up into 16 banks, allowing many different modes of operation to be resident simultaneously. Switching between them was accomplished using DIP switches. Additional paging support was included to allow the system to use the full 64K of space if necessary.

- A 64K×16 external memory using 15ns Micron MT5C2564 64K×4 SRAMs. The 'E14 provides no convenient method for accessing external memory as the intent of the processor is for all operations to use the on-chip RAM exclusively. Since the FOE board will need to gather data perhaps over large sequences of images, the on-chip store was deemed insufficient. To include external memory into the system, a hardware address pointer was added. In order to access this memory, the 'E14 writes the required address to the pointer port, and then reads the resulting output from the RAM port. Internally, the 'E14 is pipelined with an instruction pre-fetch occuring while the current instruction is being executed, and hence all external accesses cause the pipeline to stutter. This means that every external access takes two instructions cycles instead of one, and writing the pointer and reading the RAM would require 4 cycles. To reduce this overhead, the address pointer was enhanced to include auto-incrementing and auto-decrementing in hardware as well as a read-back feature. In this fashion data can be streamed into memory without needing to access the pointer on every write or read. The pointer was implemented in 7ns Lattice 22V10 pals.

- Power-on reset using a MAX701, an 80$\times$1 character LCD display for diagnostic output, as well as a keyboard and hex switches for user input to set the parameters of the system, such as the gain of the feedback loop, the value of the threshold current in the cutoff weighting function, and so on.

- The bit-selectable port on the 'E14 was used with buffering to implement the parallel interface to the host computer using the IBM 8-bit standard.

- The capture subsystem on the 'E14 handles the four system interrupt levels. One interrupt comes from the ADCs indicating a completed conversion. This is used to read in the data from the ADCs and store it in external memory. The next level of interrupt comes from the pattern generator and indicates that the current frame-pair has been completed; this is used to then update the feedback loop. The third interrupt comes from the motor controller and is used to indicate that the stage is in motion, and has finished accelerating. This is used to synchronize the feedback loop to the stage motion. Lastly, the parallel port interrupt indicates that the host computer is ready to receive another byte of output.

- 4 ADCs and 3 DACs for interfacing to and driving the FOE chip in the feedback loop.

The processing and control features of the FCP proved sufficiently flexible for most of the various configurations that became necessary during the testing of the FOE chip.

4.3.3 Interfacing the Chip to the Test System

Lastly, we turn to the ADC and DAC interfaces of the test board to the FOE chip. Figure 4-11 shows the signal flow diagram of the normal operation of the feedback loop. The four channels of processed data output from the FOE chip are the error channels e_x and e_y, the absolute value of the time derivative s_{abs}, and the squared image brightness gradient magnitude s_{quad}. These four signals are digitized by the four on-board ADCs. The conversion process is triggered by the pattern generator, and only when all four conversions are complete is an interrupt signal sent to the 'E14 in the FCP. The FCP microprocessor system, shown schematically at the bottom of the diagram, reads and stores this data, using it to update the feedback loop. The results of the feedback loop are use to drive the DACs which set the inputs to the FOE chip, namely the position voltages of the FOE estimate x_0 and y_0. The threshold in the cutoff weighting function is also controlled from the FCP by a DAC.

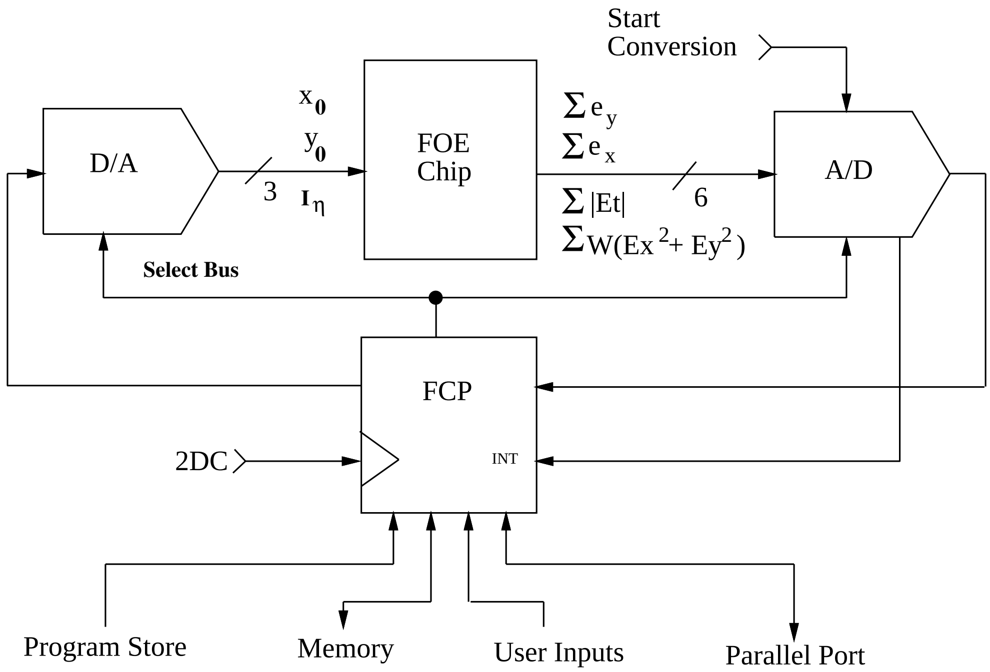


FIGURE 4-11: DSP system for implementing discrete time feedback loop.

4.3.3.1 The ADC Interfaces

The ADCs used on the FOE board are all AD7886s, a 12-bit converter available from Analog Devices, Inc.(ADI). The AD7886 track and hold amplifier acquires the input voltage signal in under 333ns, while the actual triple-flash converter takes 1μs, for an overall conversion time of 1.333μs. This converter requires an external voltage reference, and the AD586 5V reference was used. The AD7886 can be configured for unipolar input ranges of 0-5V and 0-10V, or the bipolar input range of -5V to +5V. Since the outputs from the FOE chip are all in current while we use a voltage mode ADC, conversion from current to voltage is required for each channel.

Figure 4-12 shows the interface circuit for converting the currents of the error channels to the voltage input required by the AD7886. The error currents are signed, and hence the converter is configured for the bipolar input range of -5V to 5V. The overall transfer function of the circuit is:

$$\Delta V_o = 2R_f W \Delta I_o \tag{4.2}$$

where R_f is chosen to match the output range of the error current with the input range of the ADC. The circuit functions by first converting the input differential error current to a single-ended version by mirroring one leg of the current and subtracting it from the other. The current mirror used for this purpose is a variant of a translinear circuit [58] which uses an op-amp to force matched collector currents in the bipolars. The MAT04 matched *npn*s form the basic mirror while the fet-input AD845 op-amp is used to create a virtual null at the signal current injection point. The darlington transistor biasing of the bipolar pair is done to reduce

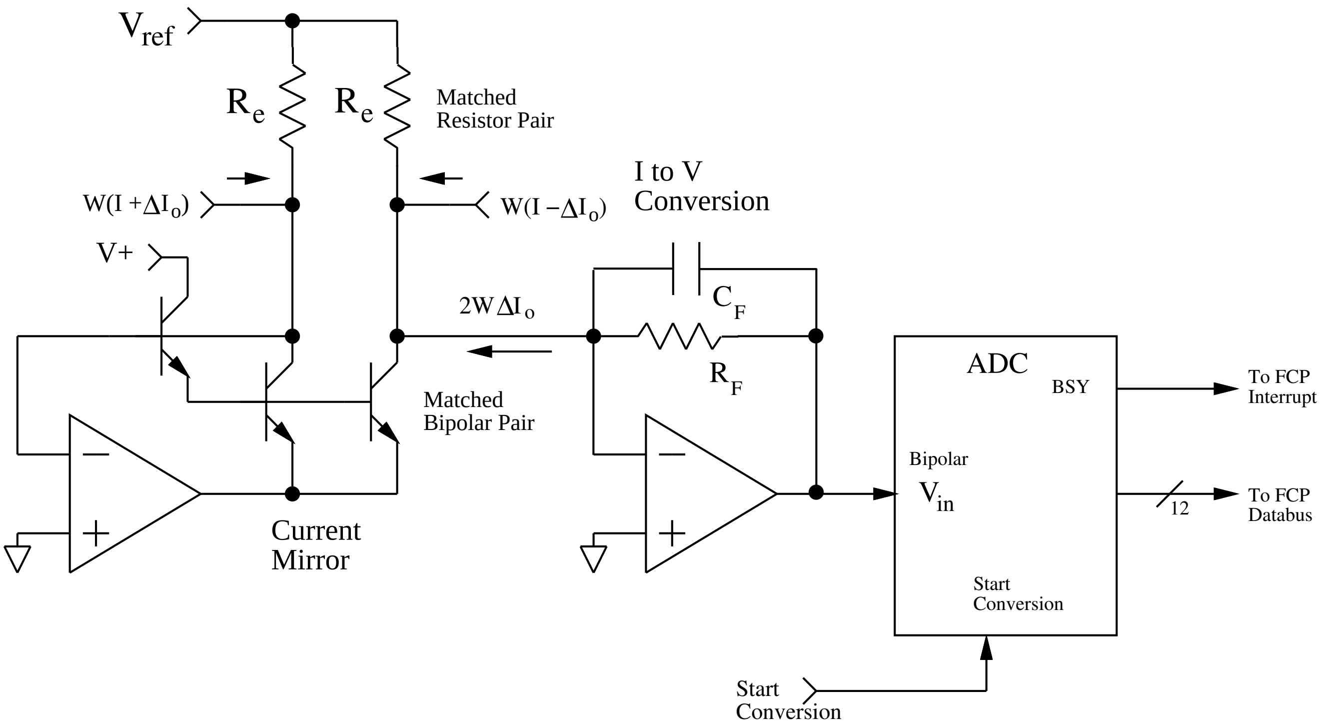


FIGURE 4-12: Board interface for the error output channels from the FOE chip.

mirroring errors due to the finite β of the transistors, while the Vishay 12-bit matched resistor pair to V_{ref} provides a constant current input to the mirror, helping to keep its bandwidth up at low signal levels. The output of the mirror is also held at a virtual null due to the output I-to-V converter [72]. Thus, each leg of the differential input current is injected into a virtual null, eliminating differential errors due to output conductance. The load capacitance at the inverting node of the output op-amp is quite large due to the lengthy distance that the signal has to travel on the board before reaching the output circuit. This long trace is due to the size of the proximal mount in which the chip is encased. The resulting large capacitance causes the feedback network around the op-amp to roll off inside the bandwidth of the op-amp, causing stability problems. Hence, compensation with a gain peaking capacitor is required in practice to stabilize the op-amp [73].

The output s_{abs} from the absolute value channel is a single-ended current flowing into n-channel devices on the FOE chip and hence the AD7886 for this channel is configured for the unipolar 0-10V range. Due to the output compliance of the n-channel transistors, the voltage must be kept above ground. Figure 4-13 shows the circuit interface, where the voltage at the injection point is kept at $V_{ref}/2$. Matched resistor pairs are once again used for accuracy, and the gain peaking capacitor C_f is for stability. The circuit results in a transfer function of

$$V_{abs} = R_a I_{abs} \tag{4.3}$$

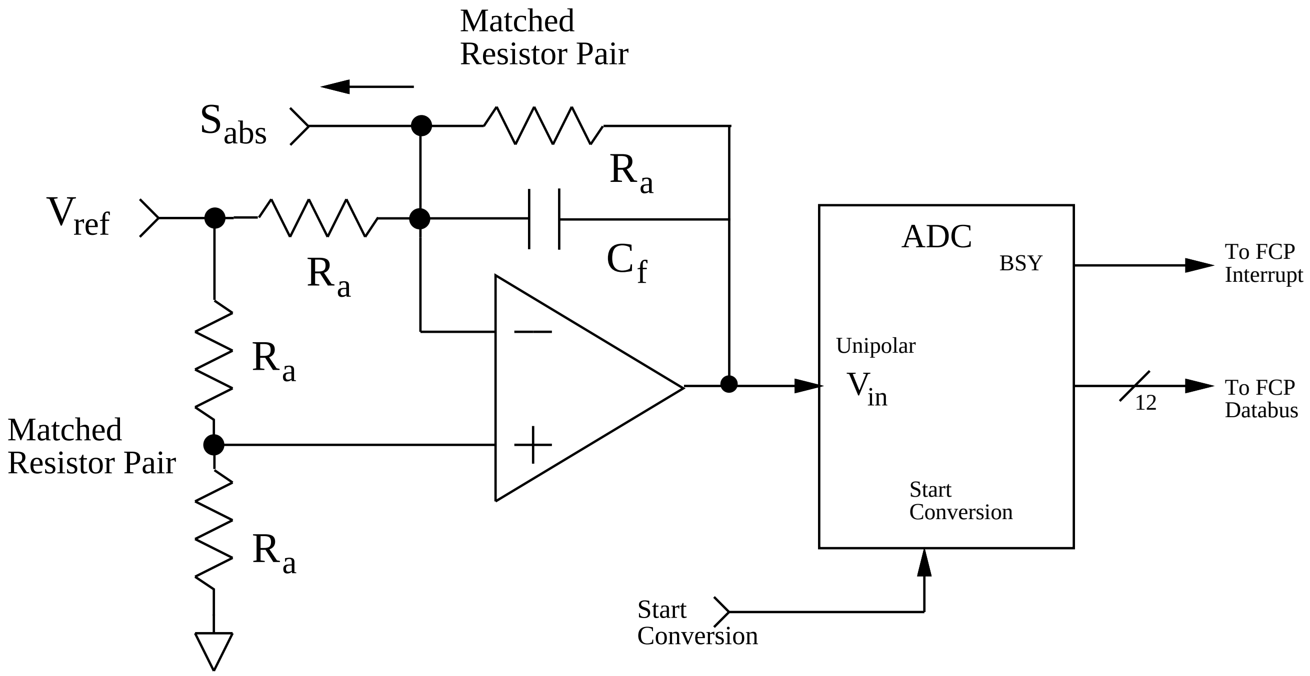


FIGURE 4-13: Board interface for the absolute value output channel from the FOE chip.

where we choose R_a to match the current output range to the voltage input range.

The output s_{quad} is also a single-ended current. However, it comes from p-channel devices and so we could have used the same simple I-to-V converter we used in the error channels. This would be entirely inverting though, and the AD7886 cannot be configured for an input range which is all negative. We could easily use another inverting stage to get rid of the minus sign, but that would entail extra components when a simpler solution is available. We instead merely shift the I/O characteristic up by the reference voltage, and configure the ADC for bipolar operation. This circuit is shown in Figure 4-14, and achieves:

$$V_{quad} = V_{ref} - R_q I_{quad} \tag{4.4}$$

With zero input, the voltage is at the converter reference; through proper choice of R_q, V_{quad} is set at $-V_{ref}$ when I_{quad} is at full scale. The offset and sign inversion are undone in software in the FCP. Note that the injection point of s_{quad} is held at $V_{ref}/2$ instead of zero now; this is well within the compliance limits of the output current.

4.3.3.2 The DAC Interfaces

The FOE chip requires three inputs for proper operation: the x_0 position voltage, y_0 position voltage, and the threshold current I_η for the cutoff weighting function. The position voltages should be constrained to be within the voltage position range used by the position encoder on the chip. To accomplish this we use the 16-bit AD7846 multiplying DAC, which has an

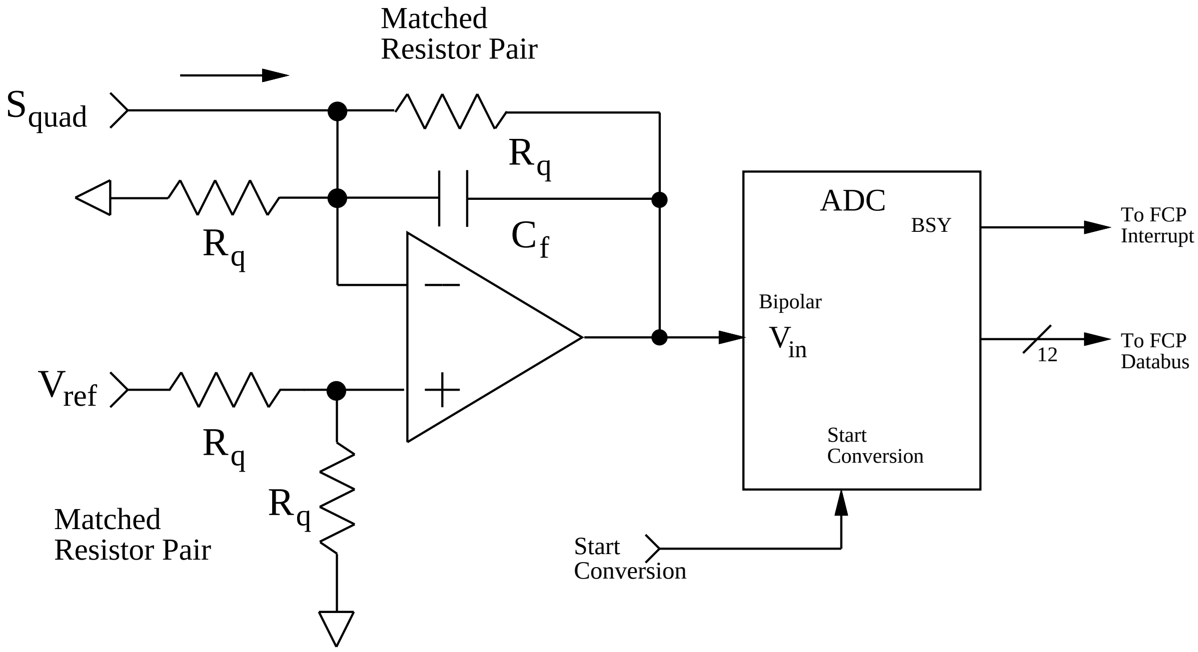


FIGURE 4-14: Board interface for the quadratic output channel from the FOE chip.

architecture shown schematically in Figure 4-15 [74]. This type of architecture uses resistor chains and switch matrices to form the output analog voltage. The key feature of this structure is that the voltage output range used by the DAC is obviously defined by the voltages placed across the resistor chain. We fix these to V_{high} and V_{low}, the same voltages used in the resistor chain in the position encoder. This naturally guarantees that the encoded position, given by the 16-bit digital word D_0, falls within the desired limits:

$$V_0 = V_{low} + (V_{high} - V_{low}) \left(\frac{D_0}{2^{16} - 1} \right) \tag{4.5}$$

The I_η current source used in the cutoff weighting function is the output of a p-channel current mirror. To bias this mirror, the FCP needs to provide a programmable current running out of the chip. The simple circuit to do this is shown in Figure 4-16 and the DAC used is the 12-bit AD7845. This DAC is configured to be inverting, and the negative output voltage is applied to an op-amp/JFET combination which converts it to a cascoded current for biasing the on-chip mirror:

$$I_\eta = \left(\frac{V_{ref}}{R_t} \right) \left(\frac{D_\eta}{2^{12} - 1} \right) \tag{4.6}$$

The $15'' \times 21''$ board for the FOE test system was designed using Cadence Design Systems' Allegro Printed Circuit Board editor and fabricated through Multek under the auspices of MOSIS. 2 signal planes and 3 internal power planes were used on the board which utilizes through-hole design exclusively for ease of debugging and modification. The internal planes form a split analog supply of ± 15 in the analog sections and a 5V digital supply in the digital

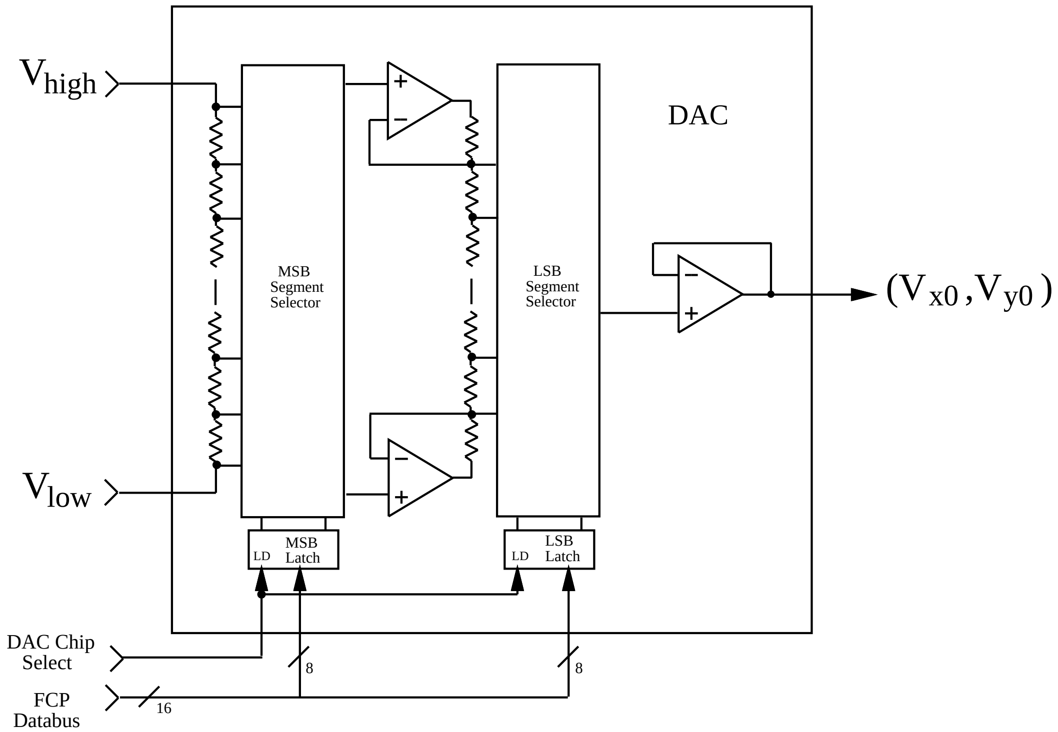


FIGURE 4-15: Monotonic voltage output DACs are used to generate the position voltages (V_{x0}, V_{y0}) for the FOE location.

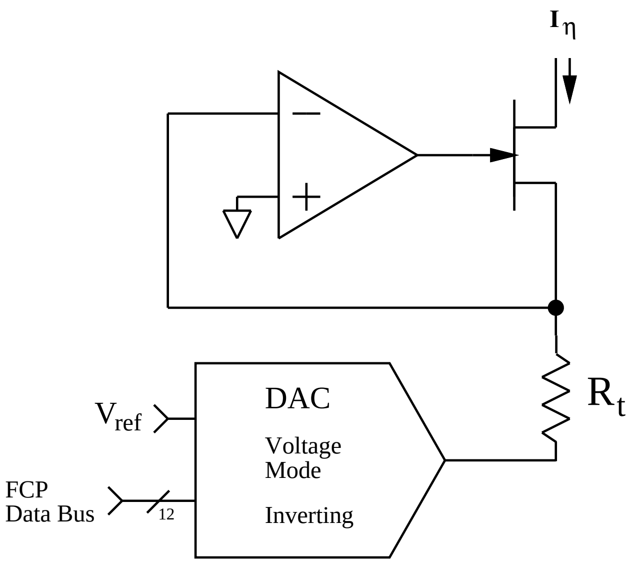


FIGURE 4-16: DAC structure for generating the threshold current I_η.

sections. Analog and digital grounds are kept separate to prevent digital returns from impacting sensitive analog nodes. This was not entirely possible in certain sections of the board, however. For example the FOE chip itself does not provide separate ground returns. Additionally, the CCD clock drivers' signal returns go through analog ground, as some of the clock levels are required to be precision levels, and hence are referenced to analog ground. To minimize the effects of this type of coupling, ground planes were used to reduce impedances in the return path, and sensitive measurements were delayed until the transient effects had dissipated. Analog and digital grounds were also tied together in the standard star configuration at the system power supply [75, 76].

4.3.4 Later Enhancements

In the course of testing the FOE chip, the FCP was constantly reconfigured to serve a variety of purposes. The flexibility of the system often was sufficient to accomplish what was required. In a few instances, however, the on-board resources needed to be extended.

The original design of the FOE system board assumed that the acquisition of raw image data through the 8 output CCD registers was handled by DSEV. However, it became necessary to find a way to use on-board resources to perform this task instead. It was decided that we would re-use the four ADCs already on the board. All that was required to do so was to perform level shifting and scaling of the CCD outputs before applying them as the inputs to the converters. Figure 4-17 shows one of the eight output channels. Buffering the floating gate are two layers of source followers. The first one is kept small in order to minimize its contribution to the load capacitance; it drives a larger transistor in the pad frame. The current source for the pad source follower is actually off-chip, and is implemented as shown using a combination of op-amp and JFET. The output of the second source follower is then buffered by a unity gain buffer and this forms the board output. Both source followers are implemented using PMOS devices to eliminate the back-gate effect which would lower the follower gain substantially. As a natural consequence of this layering of two p-followers, the voltage of the chip output during the pre-charge phase is quite high. On top of the pre-charge voltage (typically on order 6V), each follower adds a voltage shift of $V_t + \sqrt{I_{bias}/\kappa}$. Furthermore, the output follower is run at moderately high current levels in order to slew the output capacitance quickly. This capacitance is once again substantial ($\approx 50 - 60$pF) due to the long trace required to get the signal out of the proximal mount and to the output circuit. In practice, the output drive current is set such that the output voltage during pre-charge is just barely kept below V_{dd}. Of course this level cannot be allowed to exceed V_{dd} as the output pads contain diode clamps to V_{dd} and ground.

The output swing of the CCD channels is at most 1 volt. Hence, with a chip V_{dd} of 10V the signal levels we have to work with range from 10V to 9V. Of course, this must be shifted down to a more reasonable range for input to the FCP converters, as well as scaled to fit their 10V input range. Furthermore, we have 8 output channels and only 4 converters, so 2-to-1 muxing is required as well. A schematic diagram for the muxing daughter board that was constructed to allow the FCP to read raw CCD output data is shown in Figure 4-18. A variable voltage reference V_{off} is formed by the op-amp circuit to the left of the diagram. This voltage is subtracted from the input signal and injected into the main circuit. This circuit clamps the input difference to $\pm(V_d + V_z)$ where V_d is the forward drop of the Schottky diodes (≈ 0.35V) and V_z is the voltage of the bandgap zeners, which was 1.2V using LM313s. This resulted in an output clamp range of ± 1.5V. Following the clamp output, we have the required 2-to-1 muxes using ADG201HS CMOS switches and lastly an output gain stage of 10; the output of this stage is what we drive into the FCP ADCs. By varying the pot voltage in the variable voltage reference, we essentially scroll up and down a window of interest on the input waveform to which we have applied magnification. By placing the window near V_{dd}, we can fit the signal swing of the CCD output channel entirely into the appropriate input range of the ADCs. Hence, a mode of the FCP was programmed to acquire raw image data, essential for performing camera calibration and comparing system results with algorithmic ones.

Each image pair acquired and stored by the FCP in external memory requires $64 \times 64 \times 2 =$ 8Kbytes of memory. Thus the 128Kbytes of external memory in the FCP only allowed the storage of 16 frame pairs of raw data. During a motion transient, typically many more frames are required than 16, and hence the memory on the board was expanded. A memory expansion board containing 1Mbyte was constructed using the same architecture as before. Of course, now a 16-bit address pointer is insufficient and a 19-bit one was implemented instead, with the memory organized as 512K$\times$16. Since the 'E14 is a 16-bit processor, accessing the address pointer now requires two external accesses. However, the auto-increment and auto-decrement features continued to be supported, so required accesses of the pointer remain rare. With this much memory in the system, 128 image pairs (256 images) could be acquired and later transmitted through the parallel port to the host computer.

Addition of both daughter boards complicated the power supply networks substantially, and grounding proved to be a problem even though the star connection philosophy continued to be adhered to. The problem that arose was that the digital ground in the system board bounced substantially, while the memory expansion board did not bounce nearly so much. Hence, occasionally the auto-increment and auto-decrement feature would fail. The resultant failure mode was that the address pointer would occasionally increment erroneously. Through judicious placement of ground connections as well as pull-ups on critical signals, this problem was eliminated. However, when motion experiments commenced, it was observed that substantial noise from the motor system coupled into the analog sections of the FOE board, unfortunately affecting the measured data substantially with impulsive noise. Although the path of this coupling was not obvious, placing the star connection on the metal top of the optical table successfully eliminated this noise source, probably by providing a lower impedance return path.

Lastly, as we shall see in the next Chapter, the dark current of the FOE chip was, not

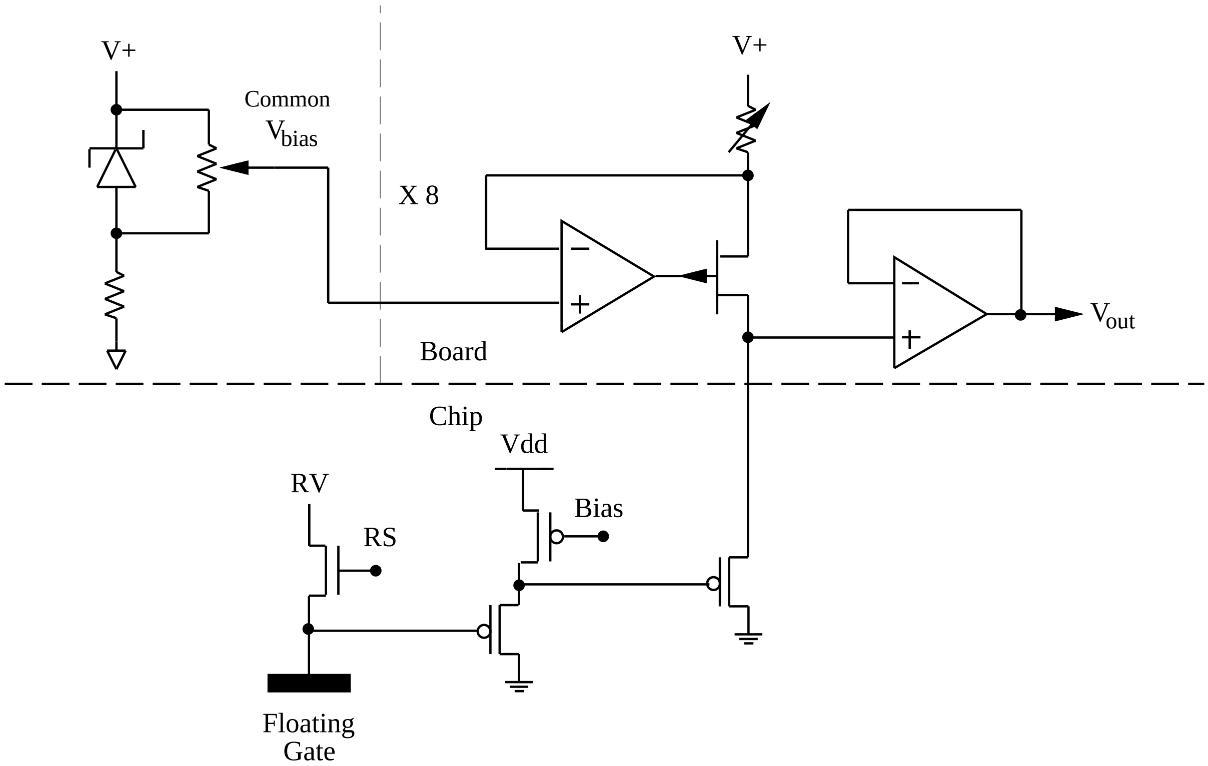


Figure 4-17: Output circuit for one of the 8 CCD I/O channels.

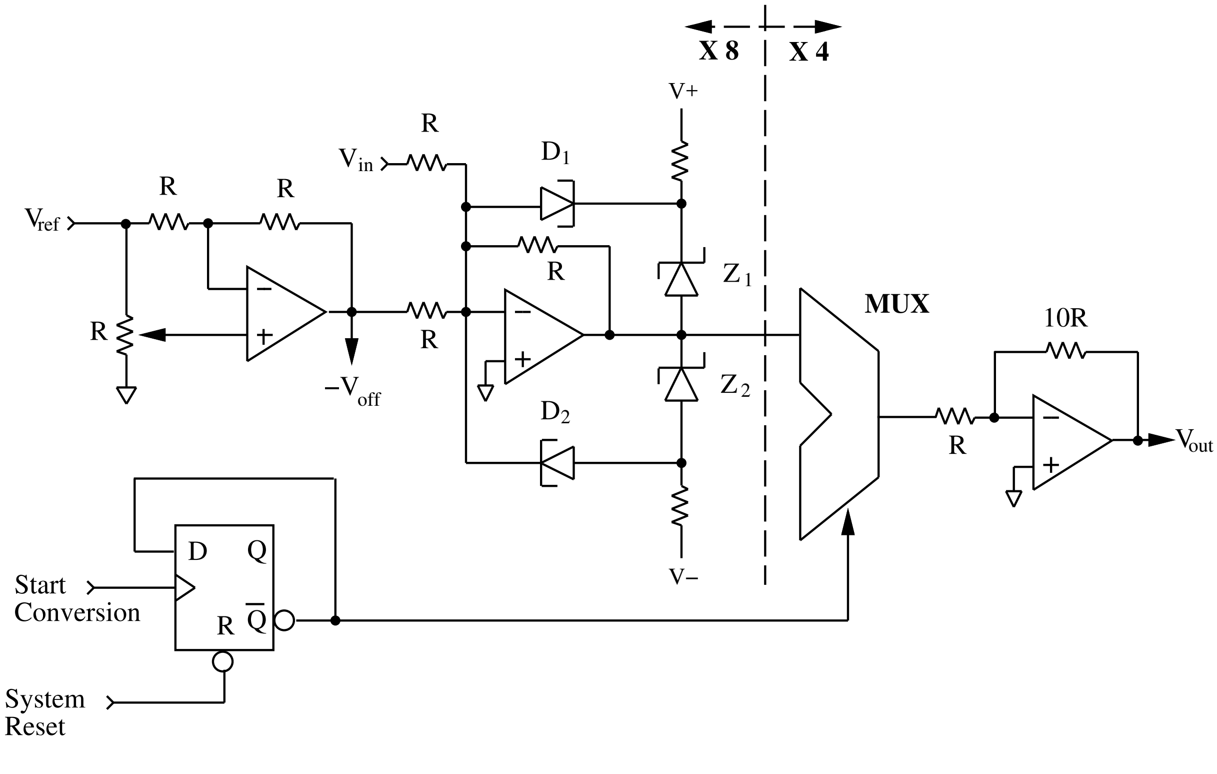


Figure 4-18: Circuit schematic for the muxing daughter board.

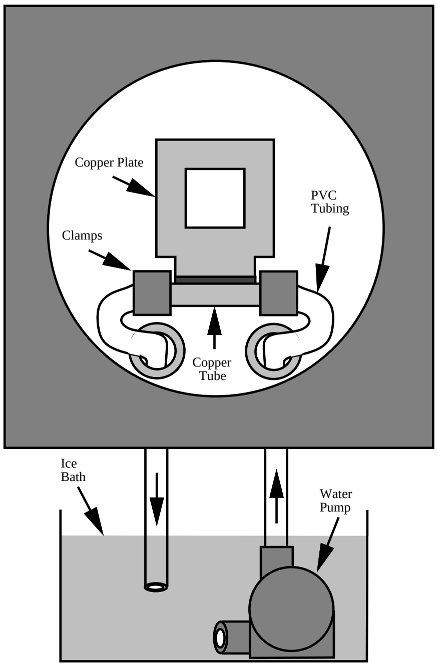


FIGURE 4-19: Simple water-based cooling system for the FOE chip.

surprisingly, substantially larger than what should be expected from a commercial CCD process. As dark current dependence is exponential in temperature, the dark current varied widely with the ambient temperature in the lab, which was poorly controlled. To remedy this situation, the test setup was enhanced by the addition of a cooling system for the FOE chip, as shown in Figure 4-19. This very simple forced cold water system consisted of an ice bath with a pump. Cold water was pumped through a copper heat sink placed in thermal contact with the chip. This sink was basically a plate placed over the chip with a thin copper tube soldered to the side of the plate. The inflow and outflow PVC tubing from the pump and ice bath were clamped to this tube at either end and exited the proximal mount through two access holes in the board as shown. In practice, this simple approach resulted in a factor of 2–3 reduction in the observed dark current, as well as reduced sensitivity to lab temperature variations.

5

The FOE Chip/Test System Results

In this chapter we present the experimental results taken from the FOE chip. The testing of the chip proceeded in three distinct stages. First, the basic functionality of the circuit structures, both CMOS and CCD, was verified. Secondly, raw image data was acquired in order for the camera calibration to be done. Motion experiments were performed, and the FOE was estimated via the raw image data on the host computer using the same algorithm as the one implemented on the chip. Lastly, the FOE chip was closed in a simple constant-gain feedback loop on the test board and the FOE estimated in real time.

5.1 Basic Functionality

5.1.1 I/O Results from the CMOS Processing Array

In order to test the basic functionality of the circuit structures on the FOE chip, a test instance of the analog processor in the CMOS array was provided. This processor is at the bottom of the array, and identical in practically every respect to those used in the actual array. However, all of its inputs and outputs were broken out separately, giving for easy access for testing. The outputs were driven into structures on the board which are identical to those serving the main array outputs. Examination of the performance of this test processor was performed first, and we discuss the results from each step in the signal flow in turn.

5.1.1.1 The Absolute Value and the Cutoff Weighting Function

At the very beginning of the signal flow in the analog processor are the four input transconductors, followed by the current mirroring that forms the differential currents representing the image brightness gradients. Access to these output currents was not provided, as this would require substantial modification of the cell, and this was avoided as the cell is intended to be representative of the processors in the array. However, the first output available off-chip is the

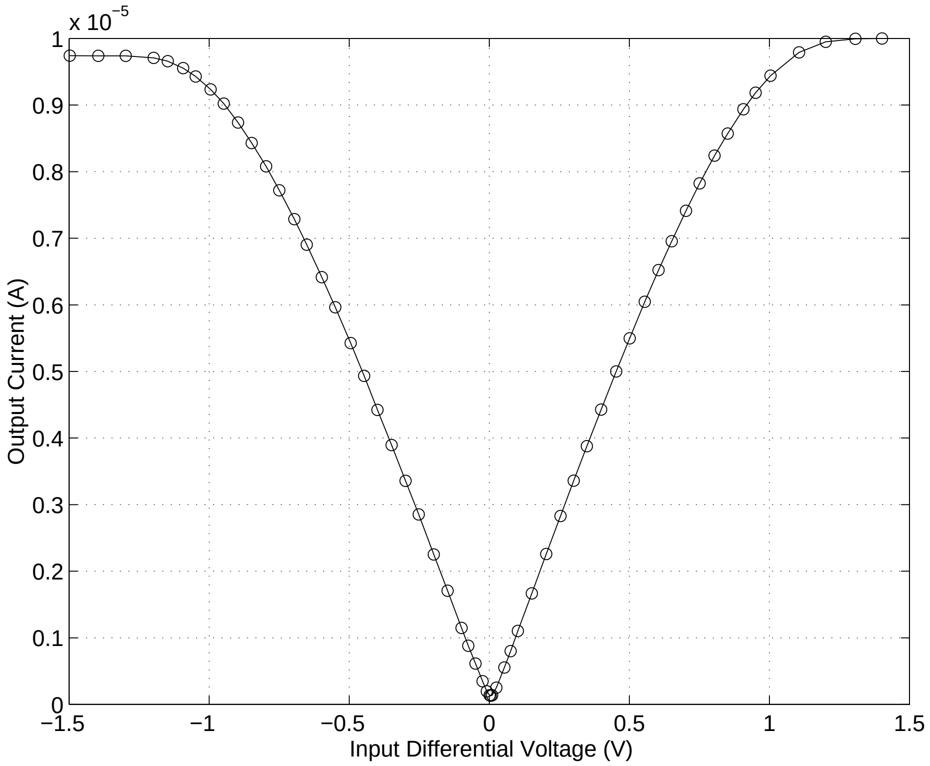


FIGURE 5-1: Absolute value characteristic measured from the test cell provided on the FOE chip.

result of absolute value structure. In order to test this structure, the 8 inputs to the transconductors are configured to drive the E_t linear combination maximally. Of course, since the three partials are formed from orthogonal linear combinations, the other two are zero under this condition. Figure 5-1 shows the measured absolute value characteristic. The nominal input bias current for the processor was set at 5μA, and adjusted until the maximum output current was $\approx 10\mu$A. This bias will remain the same for the rest of this discussion. The input/output curve shows the desired input range of about 1V and output range of 10μA under these conditions.

A copy of this signal is also used internally in the cutoff weighting function. Unlike the processors in the main array, the output of the latch in the test cell weighting function is driven off-chip by two inverters, and hence we can examine the result of the weighting decision. In order to avoid having the inverters introduce unbalanced capacitive loads, a dummy inverter is placed on the side of the latch that is not driven off-chip. Additionally, the switch point of the inverters was placed so that the pre-charge voltage used on the latch was not in the inverter's high gain region.

To test the cutoff weighting function, the threshold current I_η was set and E_t was again strobed maximally. The input voltages where the output switched high 50% of the time were noted. Since the weighting function is even, there is both a positive and a negative switch point. Both of these points are shown as a function of threshold current I_η in Figure 5-2 and exhibit the

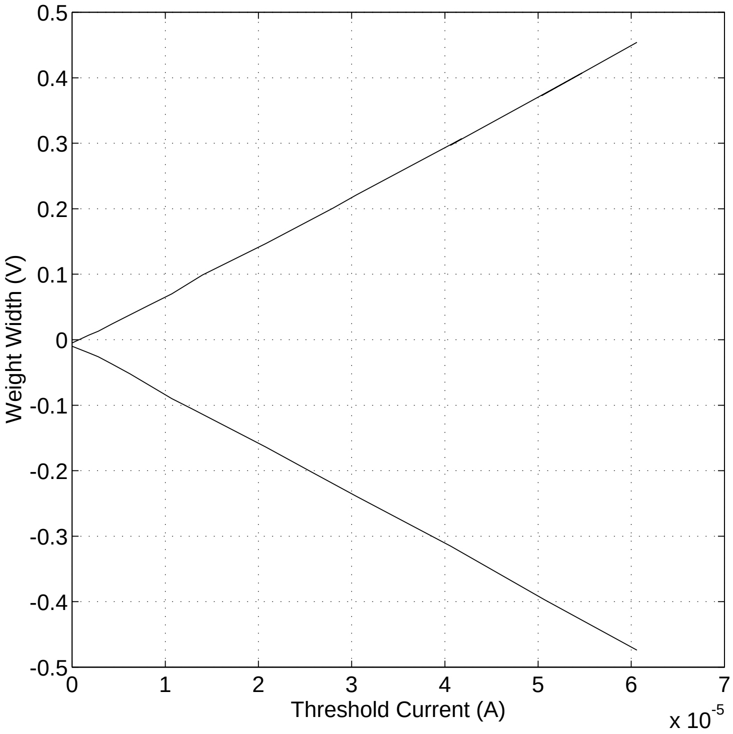


Figure 5-2: The cutoff weighting function characteristic, large scale. E_t is driven maximally and the lines represent the 50% switching points.

desired linear dependence. Below the upper line and above the lower line, the weighting function evaluates true, while above the upper line and below the lower line the function evaluates false as required. The FOE algorithm uses this function to isolate stationary points, and smaller function widths are therefore desirable. Figure 5-3 shows a magnification of the region near the origin. Clearly, the characteristic displays an offset on the order of 0.5%, which will limit our minimum sensitivity. Algorithmic simulations indicate that weighting widths of several percent are necessary for adequate system performance, and this offset is close enough to this regime to be of some concern.

5.1.1.2 Estimating the Gradient Magnitude

The differential currents representing the brightness gradients E_x and E_y are input to the quadratic current squarer, and a copy of this signal is sent off-chip. To test this output, the inputs to the transconductors were configured to drive E_x maximally and of course, $E_y = 0$ under this condition. The measured output current from the gradient magnitude channel is shown in Figure 5-4. The output displays the expected quadratic behavior, except for a rather large offset. An offset for zero input is a well known problem with squaring circuits, and provisions were made in the design to cancel the offset in this channel. However, the cancelation

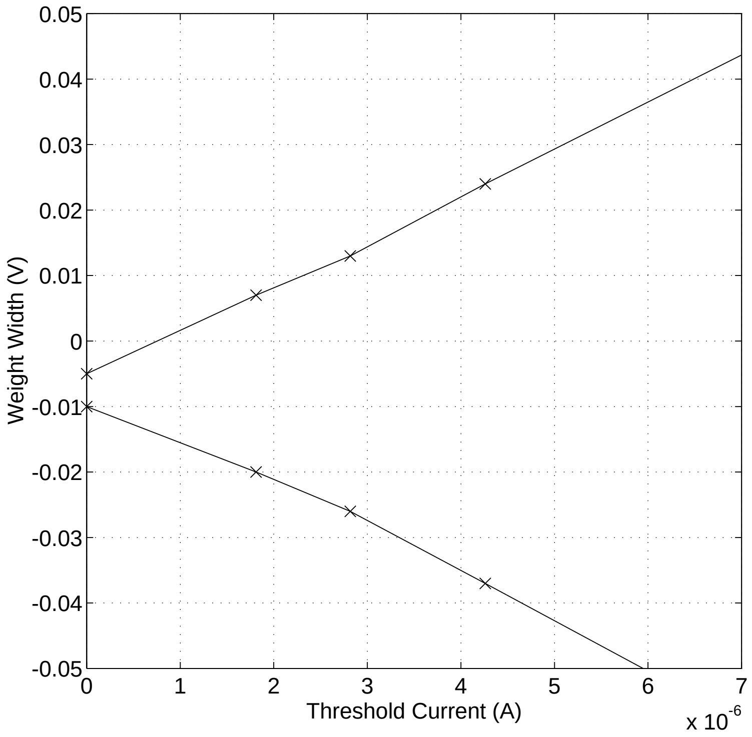


FIGURE 5-3: The cutoff weighting function characteristic, small scale.

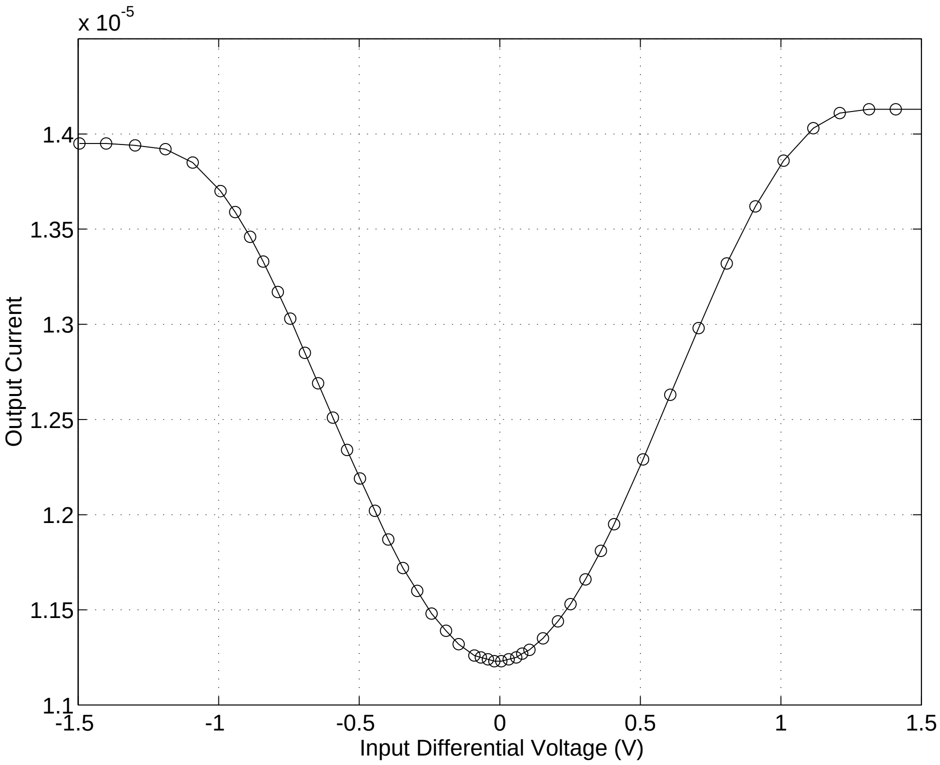


FIGURE 5-4: Measured quadratic circuit characteristic with E_x driven maximally.

was insufficient. This is the first substantial layout error discovered on the FOE chip. In the biasing network, a copy of the quadratic circuit is driven with a balanced differential current derived from the same bias as the transconductors. This current is mirrored and subtracted from the quadratic circuits in the main array. Unfortunately, the main array sums the contributions from two such circuits (one for E_x^2 and one for E_y^2) to form $E_x^2 + E_y^2$; the biasing therefore provides cancelation of only 1/2 of the offset because the mirror into the array is unity gain as opposed to having the required gain of two.

Note that this offset is substantially larger than the actual signal current, and this is a problem for the output circuits which are scaled to match this signal. Furthermore, because of the weighting function, this offset in the main array is in fact now variable, depending on the results of the decision. To compensate for these effects, the gain on the quadratic channel had to be lowered to avoid saturating the output circuits. Additionally, the system will now need to have a special cycle in order to cancel this offset in software. This cycle will occur after the weighting decision has been made; we merely pre-charge the floating gate amplifiers, effectively zeroing out the inputs to the transconductors. The measured output currents from the chip during this time will then be the offsets modulated by the last weighting function decision.

5.1.1.3 The Multiplier Core

We now turn to the multiplier core which provides the error channel outputs. No intermediate product is provided; only the output of the cascade of the two sets of multipliers goes off-chip from the test cell, mimicing the output of the main array. To test this structure, we drive $x - x0$ and E_x maximally. Again E_y is zero and we can further set $y = y_0$, leaving only $x - x_0$ and E_x dependence in the x-channel output. Under this drive, E_t is also zero; additionally, we set I_η maximally to guarantee that the weighting function always switches true. Under these conditions, we find that the x-error becomes:

$$e_x = W(E_t, \eta)E_x\Big((x - x_0)E_x + (y - y_0)E_y\Big) = (x - x_0)E_x^2 \tag{5.1}$$

and hence we have a linear dependence on $x - x_0$ and a quadratic dependence on E_x. In Figure 5-5, we show the measured family of quadratic curves parameterized by $x - x_0$, where E_x is taken as the ordinate. In Figure 5-6 we swap the role of the inputs, showing the linear behavior with respect to $x - x_0$. These curves match well the behavior predicted by our normalized models in Figures 3-42 and 3-43.

The behavior of the y-channel is qualitatively the same as the x-channel, as the entire circuit is symmetrical with respect to the two channels. In the $x - y$ scope trace in Figure 5-7, the inputs of the transconductor were configured to drive E_y maximally, and a 1kHz sine-wave was used as an excitation. $y - y_0$ was set as a DC voltage, and the resulting output current sensed through a 7.8kΩ resistor.

Lastly, the settling time of the analog processor needs to be small enough to operate at the

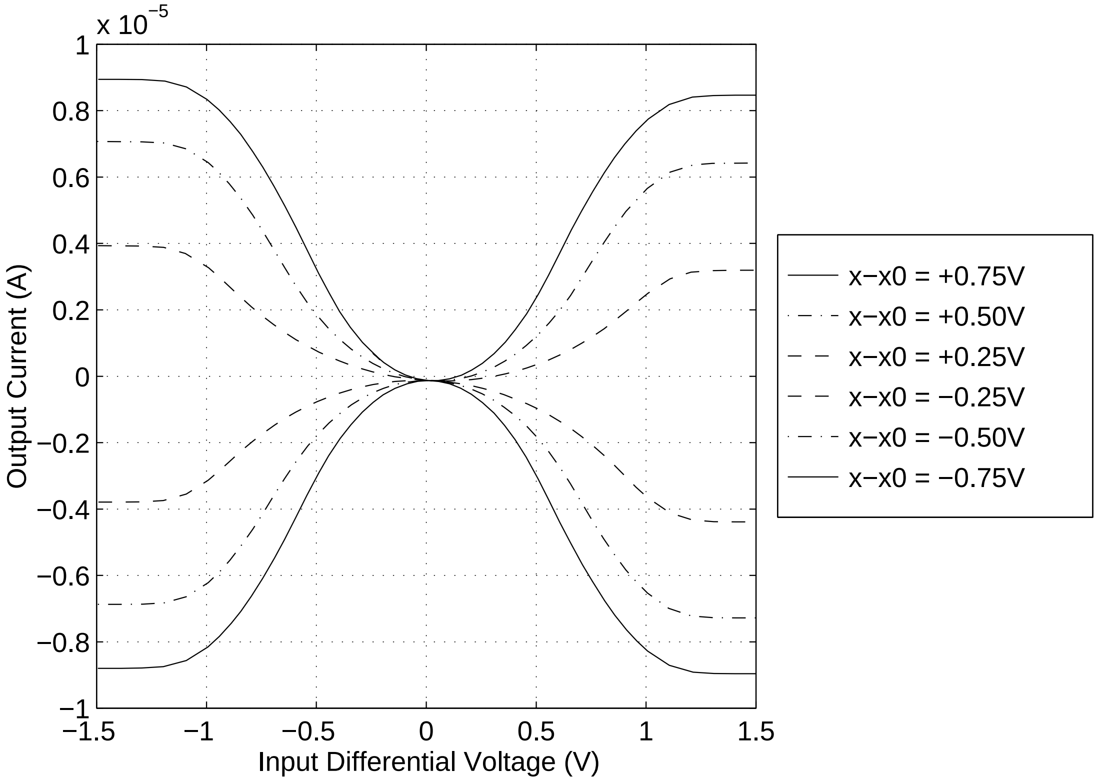


FIGURE 5-5: The family of curves for the x-channel output current of the test cell parameterized by $x - x_0$.

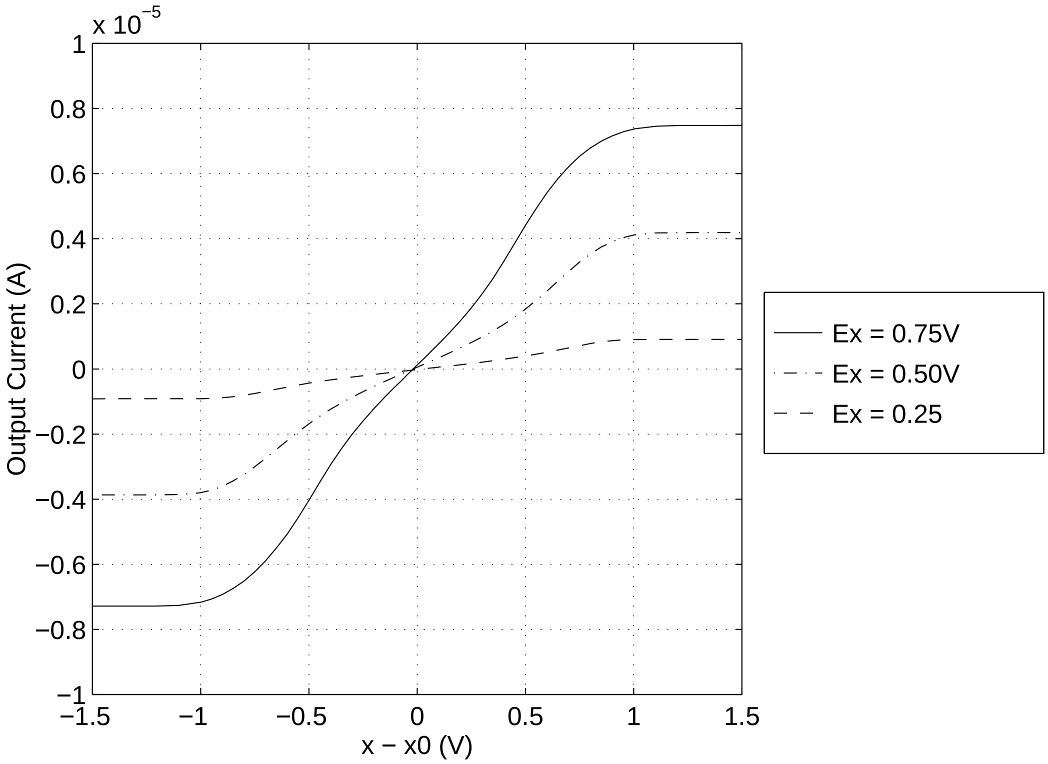


FIGURE 5-6: The family of curves for the x-channel output current of the test cell parameterized by E_x.

target 1kHz frame rate. At this speed, we have $\approx 15\mu$sec to perform all the required operations: shifting image charge right twice with a charge sense on the second, performing the weighting function operation, measuring the output, pre-charging the floating gate amplifiers again and finally measuring the offset. As a conservative specification, the processor is required to settle in less than 10% of the allowed column processing time. Figure 5-8 shows a scope photograph of the processor meeting this specification. The top trace is the latch reset clock ϕ_{reset}, the second trace is the latch clock ϕ_{latch}, the third trace is the output signal from the x-channel, and the fourth trace is the weighting function decision $\bar{W}$. The reset voltage is set so that during the pre-charge phase, output current from the main channel is cutoff ($\bar{W}$ is high) using the cascode transistors in Figure 3-33, and hence the resulting output voltage during reset is zero. When ϕ_{latch} goes high, the latch flips and now $\bar{W}$ evaluates low, and clearly the output signal settles in under the required 1.5μsec.

5.1.1.4 The Position Encoder

The last of the CMOS structures on the chip is the position encoder. The encoder was designed to provide 100mV/processor position encoding, or a maximum range from high to low of 6.3V. The first layer of multipliers in the analog processor was designed to accommodate this, with an allowed input range of 1V to 8V. Figure 5-9 shows x-position output strobe of the position encoder. V_{high} was set at 7.4V, while V_{low} was set at 1V. The image acquisition time in this figure was set at 0.5 msec, twice as fast as necessary. Figure 5-10 shows a blowup of the strobe near the high end, demonstrating the desired 100mV/step. The settling time given by simulation (see Figure 3-48) was $< 1\mu$sec; actual performance appears to verify this.

5.1.2 Basic CCD results

In this section, we quantify the basic operating parameters of the CCD devices in the FOE chip. There are four basic parameters of interest. The first parameter is the sensitivity of the floating gate amplifiers, along with a measure of their nonlinearity. The second parameter of interest is the transfer efficiency, quantifying how well we can move charge about the CCD structures. The third parameter is the dark current, which gives the amount of thermally generated background signal present. Lastly, the quantum efficiency of the imager is measured, indicating how efficiently our collection of photon induced signal is performed. For the first three measurements, we use the I/O shift registers as test vehicles. The last measurement, of course, requires using the imaging array.

5.1.2.1 Charge Input and Output

We first turn to examining the floating gate amplifiers because all of our basic measurement information is taken with these structures. The input/output shift registers have output struc-

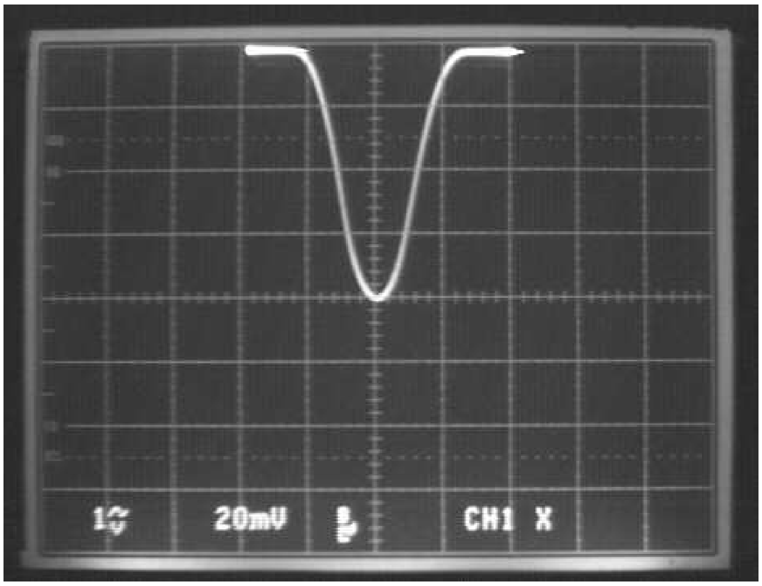


FIGURE 5-7: Scope trace of y-channel output with E_y maximally driven.

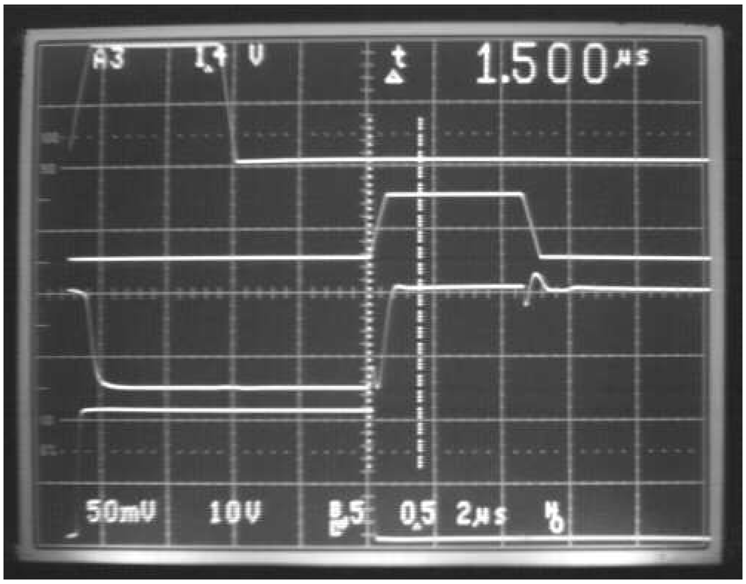


FIGURE 5-8: Scope photograph showing the settling time of the test cell. The output circuit is the same as on the main array. The traces in order from top to bottom are ϕ_{reset}, ϕ_{latch}, e_x, and $\bar{W}$.

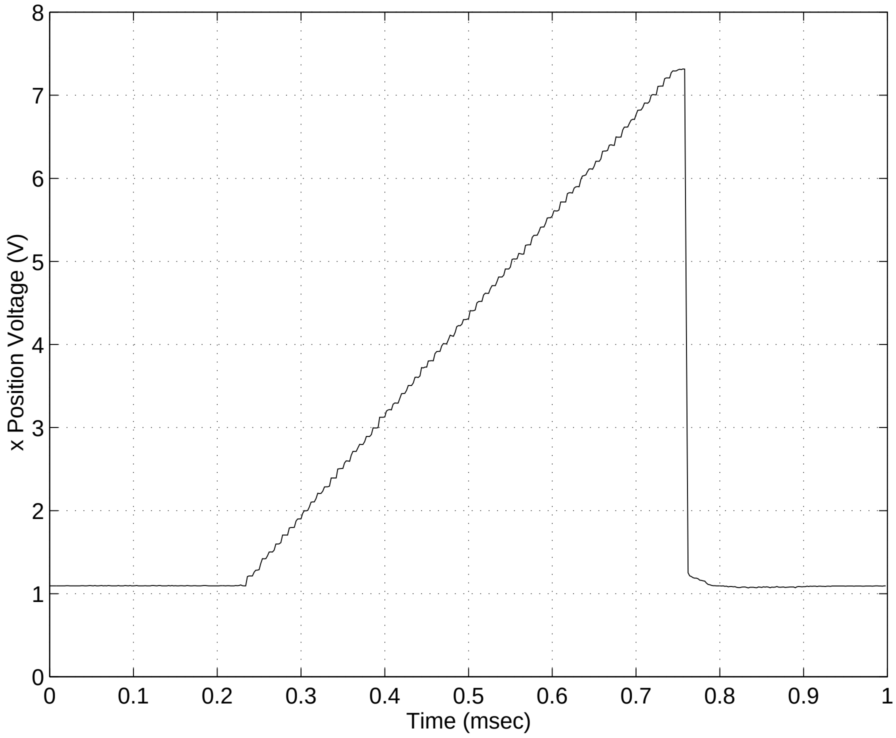


FIGURE 5-9: Scope waveform showing operation of the position encoder over an image frame.

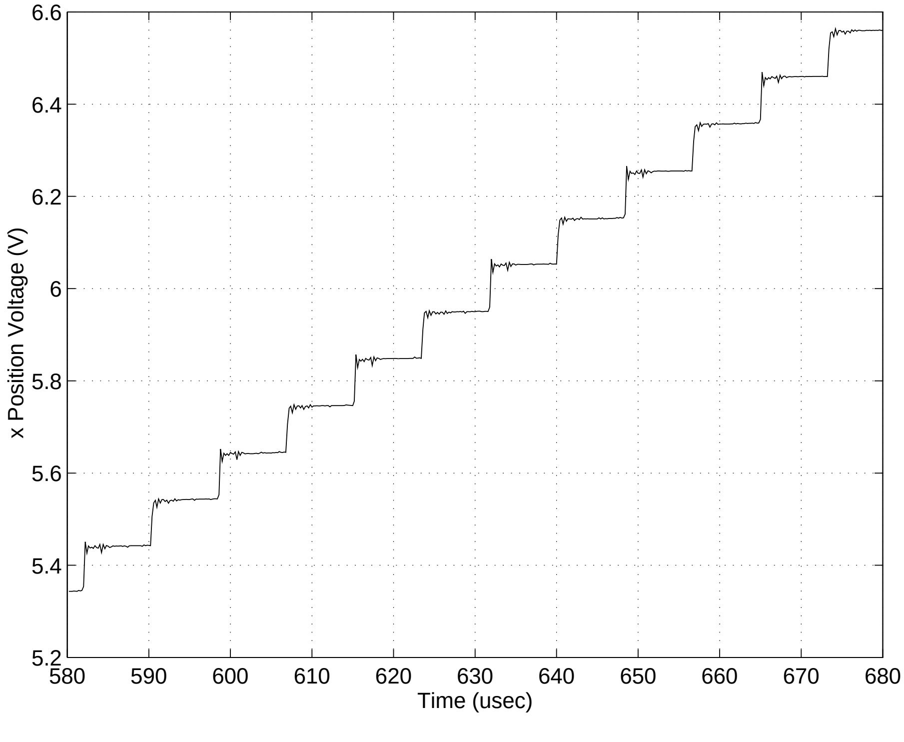


FIGURE 5-10: Magnified scope waveform showing operation of the position encoder.

	Floating Gates
Gate size:	30μm $\times$ 15 μm
C_{ox}:	$450\mu\text{m}^2 \times 0.86$ fF/μm^2 = 387 fF
	Reset Transistor
Size:	8μm $\times$ 7 μm
Junction:	$40\mu\text{m}^2 \times 0.13$ fF/μm^2 = 5.2 fF
Sidewall:	26μm $\times 0.53$ fF/μm = 13.8 fF
Overlap:	8μm $\times 0.29$ fF/μm = 2.3 fF
Total:	21.3 fF
	Source Follower
Size:	8μm $\times$ 7 μm
Gate:	$56\mu\text{m}^2 \times 0.86$ fF/μm^2 = 48.2 fF
Overlap :	8μm $\times$ 0.29 fF/μm = 2.3 fF
Total:	50.5 fF
	Transfer Gates
Overlap size:	36μm $\times$ 1.2 μm
C_{ol}:	$43.2\mu\text{m}^2 \times 0.43$ fF/μm^2 = 18.6 fF

Table 5-1: Table of device parameters for the floating gate amplifiers.

tures which are identical to the ones in array feeding the analog processors, so we may use their parameters as a measure of the characteristics of the devices in the main array. Figure 3-25 showed the lumped circuit model for the buried channel CCD structure and in this model there are four relevant capacitances: C_{ox}, C_l, C_{d1}, and C_{d2}. C_{ox} is simply the oxide capacitance from the semiconductor surface to the gate. For the particular geometry used in the FOE chip, Table 5-1 indicates that $C_{ox} = 387$ fF.

C_l is the load capacitance seen by the gate consisting of the components from the reset transistor, the input capacitance of the source follower, and the overlap capacitance between the floating gate and adjacent phases. The input capacitance of the source follower is reduced by $(1-A)$ where A is the gain of the follower. Typically, the great majority of the gain reduction in a source follower is due to the back-gate effect. In the FOE design, we use p-channel devices exclusively in the followers; this lets us eliminate the variation in threshold due to the back-gate and consequently the gain should be quite close to unity. By pre-charging the floating gate and measuring the output as a function of pre-charge voltage, we can measure the response of the source follower to the output. The resulting characteristic is shown in Figure 5-11. In driving the chip output, two source followers (both p-type) were employed, and they each give about 2V of output shift. The resulting gain from the cascade is 0.97, very nearly unity. The nonlinearity in the source follower characteristic is shown in Figure 5-12 indicating less than 0.2% total nonlinearity. Referring to Table 5-1, which also lists all of the various contributions

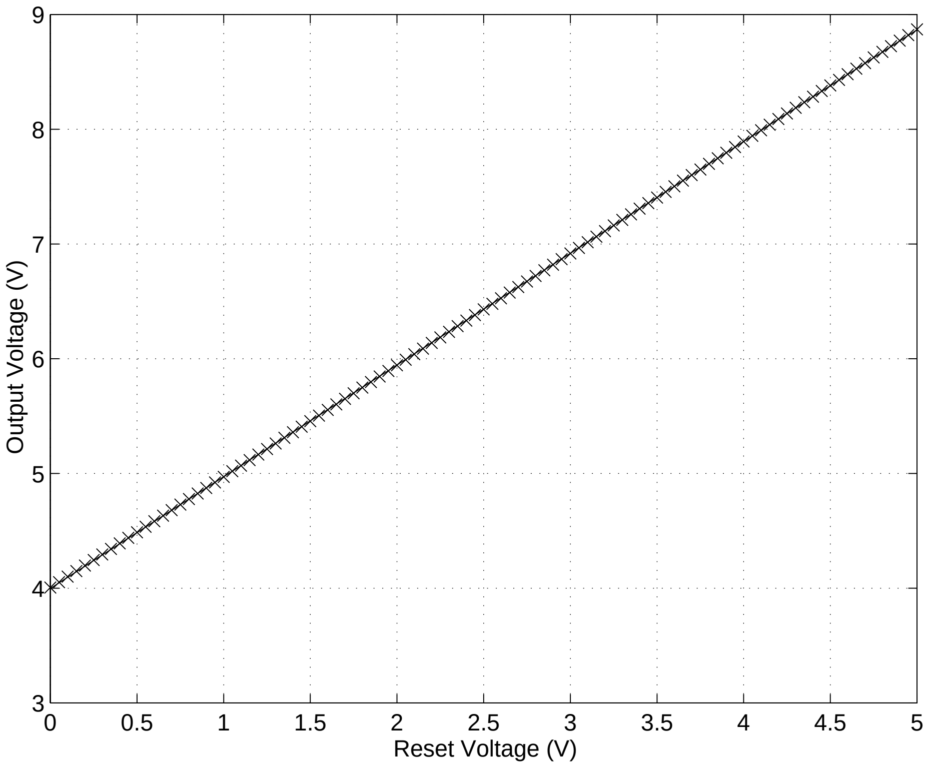


FIGURE 5-11: Source follower characteristic from pre-charge voltage to output.

to C_l, we find that:

$$C_l = 21.3 \text{ fF} + (1 - 0.97) \times 50.5 \text{ fF} + 2 \times 18.6 \text{ fF} = 60.0 \text{ fF} \tag{5.2}$$

About half of this capacitance is due to the overlap with adjacent clock phases. Design rules for the Orbit process require 2μm minimum overlap between adjacent gates in layout; as this process uses an over-etching technique on poly2, the resulting overlap is $\approx 1.2\mu$m. Inter-poly spacing is controlled to twice the gate oxide thickness, and these two facts yield rather high parasitic overlap capacitances.

C_{d1} is the depletion capacitance from the channel to the oxide, and C_{d2} is the depletion capacitance from the channel to the substrate. Calculation of these capacitances is complicated by the presence of signal charge in the CCD channel. As more charge is added, the width of the channel increases spreads, which in turn decreases the width of the depletion regions, resulting in an increase of C_{d1} and C_{d2}. To find an expression for this spreading, we can assuming charge neutrality in the channel region. This gives the width of the channel as [7]:

$$W_{ch} = \frac{Q_n}{qN_d} \tag{5.3}$$

Using a charge packet of 1×10^6 e-, a doping concentration of $N_d = 4 \times 10^{16}/\text{cm}^3$, and a gate area of $30\mu\text{m}\times15\mu$m we find that the channel width is 0.056μm. If we assume that this width

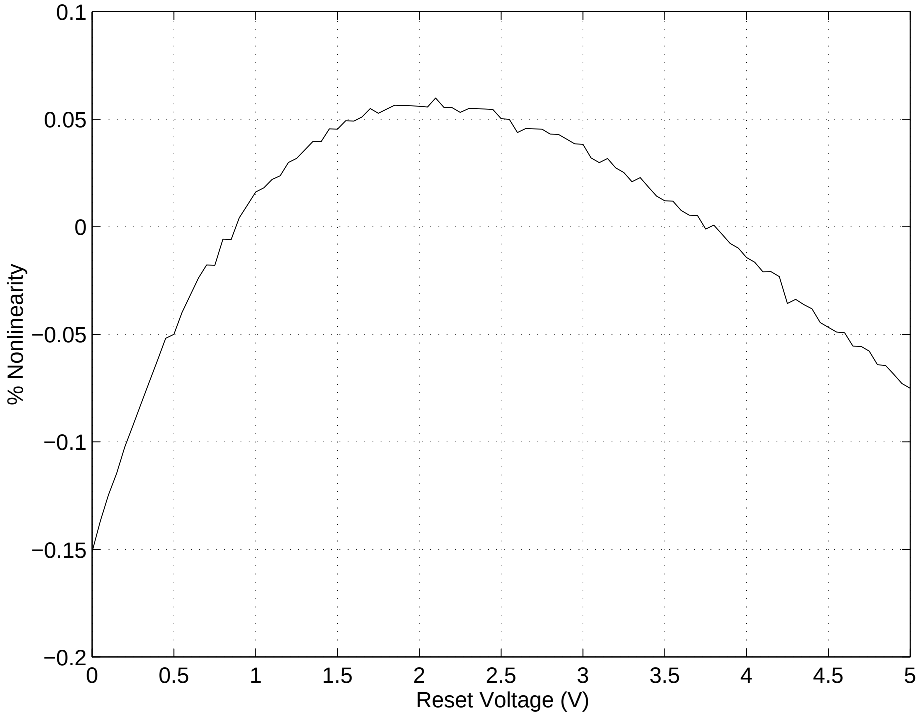


FIGURE 5-12: Nonlinearity of the source follower characteristic.

splits evenly amongst the two regions [11], then we can approximate C_{d1} by

$$C_{d1} = \frac{\epsilon_{si} W L}{x_{\max} - W_{ch}/2} \tag{5.4}$$

For the Orbit CCD process, $x_{\max} = 0.4\mu$m, and this results in a capacitance $C_{d1} = 126$ fF. With no charge in the channel, there is no spreading and we have $C_{d1} = 117$ fF, a change of $\approx 7\%$. As a conservative worst case estimate, we can assume that all of the spreading occurs in C_{d1}; this would result in a change of $\approx 15\%$ from 0 to 1Me- channel charge.

The C_{d2} capacitance depends on the channel to bulk depletion width. Typically, the diodes in the input structures on the FOE chip are kept at 10V, and hence the depletion region of C_{d2} is quite large. With a substrate doping of $1 \times 10^{15}/\text{cm}^3$, we find that the depletion width is 3.6μm and hence $C_{d2} = 13$ fF. The impact of the presence of channel charge on C_{d2} is $\approx 0.056/3.6 = 1.6\%$.

Now that all four capacitances have been accounted for, we can now calculate the sensitivity of our floating gate amplifiers using the output equation:

$$\Delta V_{out} = \frac{Q_s}{C_l} \left(\frac{\frac{1}{C_{d2}}}{\frac{1}{C_{d1}} + \frac{1}{C_{d2}} + \frac{1}{C_l} + \frac{1}{C_{ox}}} \right) \tag{5.5}$$

With $C_l = 60$ fF, $C_{ox} = 387$fF, $C_{d1} = 126$ fF and $C_{d2} = 13$ fF, we find a sensitivity of $\approx 2\mu$V/e-. Note that in our case $C_{d1} > C_l$, and hence if we approximate away both C_{ox} and C_{d1} in the

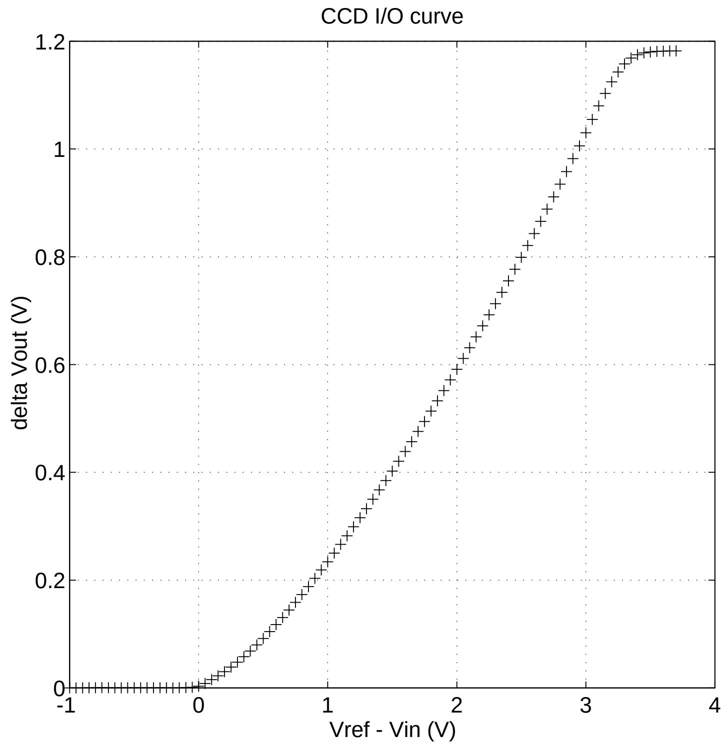


FIGURE 5-13: Input to output characteristic of the I/O shift register.

expression, we find:

$$\Delta V_{out} \approx \frac{Q_s}{C_l}\left(\frac{\frac{1}{C_{d2}}}{\frac{1}{C_l}+\frac{1}{C_{d2}}}\right) = \frac{Q_s}{C_l}\left(\frac{1}{1+\frac{C_{d2}}{C_l}}\right) \tag{5.6}$$

Clearly, the 1.6% variation of C_{d2} is reduced by the ratio of C_{d2}/C_l resulting in an overall nonlinearity in the output of at most 0.35%, which is on a par with the nonlinearity of the cascade of source followers buffering the output.

For a pre-charge voltage 4V above the DC blocking gate, a maximum output swing of 1.2V is observed in practice, corresponding to maximum charge packet of 600,000 electrons. Charge packets larger than this size cause the floating gate to overflow, and the sensitivity is substantially reduced as the excess charge fills in the blocking gate.

The input structures on the I/O shift registers are of the split-gate type as shown in Figure 3-15b. The resultant input to output characteristic is shown in Figure 5-13. This curve shows a substantial amount of nonlinearity, which could come from the input structure, the output structure, or both. However, when we later discuss responsivity to light, we find a quite linear relationship. As a result, we can conclude that the nonlinearity is mostly due to the split-gate input. Inputting charge using buried channel input structures is typically fairly nonlinear [11]; an improvement in the FOE chip would involve the addition of surface channel devices for input, followed by buried channel devices for the rest of the array. Current Orbit de-

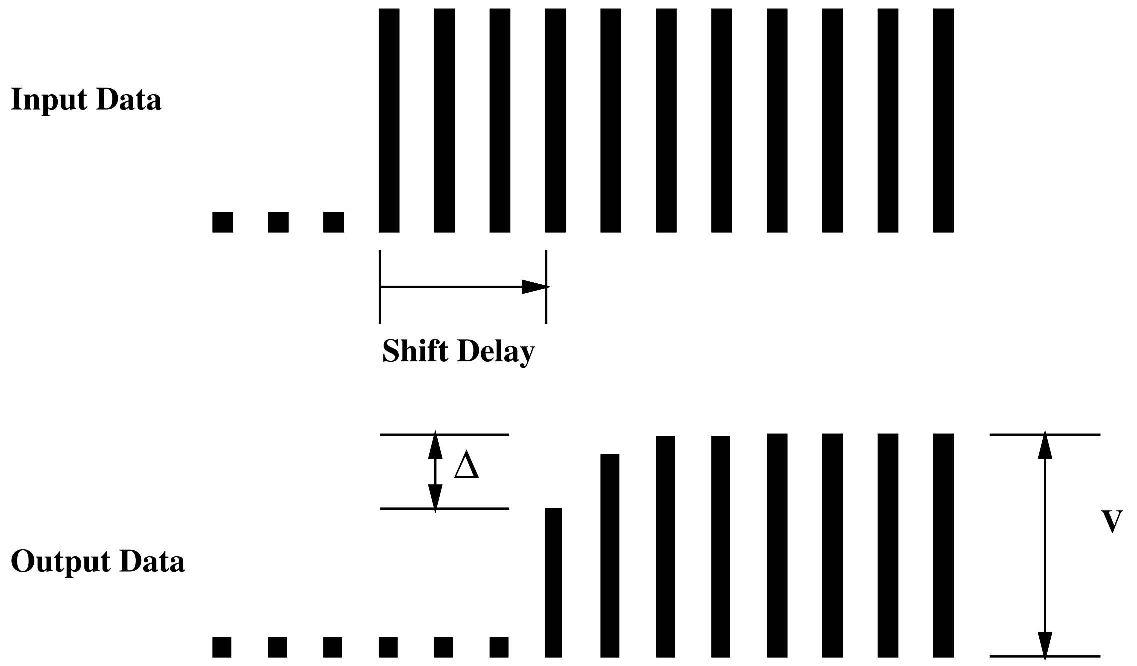


FIGURE 5-14: Illustration of the measurement of CCD transfer inefficiency using the step response of a shift register.

sign rules do not allow surface channel devices, however with minimal modification this should be possible to include on the FOE chip. The nonlinearity of the input does not substantially impact the performance of the chip as the structure is included for testing purposes only.

Input to output sensitivity is overall about 1 volt of output for 3 volts of input. This leads to an input sensitivity of about 200,000 e-/V at the input. The disparity of input sensitivity compared with output sensitivity is due to the reduced size of the floating diffusion compared with a standard gate size in the array.

Lastly, the sharp cutoff at 1.2V has been intentionally placed in this measurement. Physically, right after the floating gate in the register is a ϕ_4 gate followed by a reverse biased diode forming a charge dump. Typically, after the floating gate is filled, the charge back-spills into the DC blocking gate ϕ_2, as this potential level is lower than the off-level of ϕ_4. This results in a kink in the curve, and the output once again becomes linear, but at a reduced slope. In this measurement, the off-level potential of ϕ_4 was placed low enough so that the excess charge spills over into output diode instead of the blocking gate, resulting in the sharp cutoff.

5.1.2.2 Transfer Inefficiency

The method we use for measuring the transfer efficiency of a CCD is to measure the step response of a shift register [44]. The basic concept is shown in Figure 5-14. Initially, the input is zero and hence so is the output. At some later time, the input is stepped to a constant input value; after a number of shifts equal to the length of the register, the information appears at the output. The first charge packet is reduced by the inefficiency of each gate it went through and subsequent charge packets gain in amplitude by picking up the charge left behind by the preceeding packet. Eventually, a steady-state is achieved in which amount of charge lost by a packet is equal to the amount left behind by the preceeding packet. The output under this condition corresponds to an unattenuated input.

If we turn off the input, the last charge packet out of the register is attenuated relative to steady state by the inefficiency of all the transfers it went through before reaching the output. It is reduced by:

$$V_o = (1 - \epsilon)^N V_{ss} \tag{5.7}$$

where ϵ is the transfer efficiency per gate, N is the number of gates in the register, and V_{ss} is the steady state output voltage. Taking the difference between V_{ss} and the last packet out of the I/O register V_o and dividing by V_{ss}, we have:

$$\frac{\Delta}{V_{ss}} = \frac{V_{ss} - V_o}{V_{ss}} \approx N\epsilon \tag{5.8}$$

The difference between the steady state packet and the output packet is quite small. For this measurement, we used a variant of the clamp shown in Figure 4-18, wherein the differential amplifier had a gain of 5.08 and in which Schottky diodes were placed in both directions about the op-amp instead of the zener structure, leading to an output voltage range of $V_d = \pm 0.35$V. In this configuration, the Δ was measured to be no more than 20mV with 57 gates of transfer at shift rates of 2μsec. The resulting inefficiency is:

$$\epsilon < \frac{20mV/5.08}{57 \times 1V} = 6.9 \times 10^{-5} \tag{5.9}$$

The efficiency is therefore > 0.99993 while the MOSIS specification for this parameter is 0.99999. Our measurement is at the limit of the resolution of our test setup, but is certainly adequate for operation. Since the processing mode of the chip is column parallel, the worst case charge packet goes through at most 530 transfers, and this results in at most 3.65% signal loss. Our system is not dependent on absolute levels, however. Only local gradients are relevant to the algorithm, and hence differences between the charge packets in the four output amplifiers in a row are only due to at most 16 shifts resulting in a difference of only 0.11 %.

5.1.2.3 Dark Current

Thermally generated charge is constantly being collected by the CCD devices during normal operation. The figure of merit for this dark current is the current density J_D, and by optimizing fabrication to minimize dark current, current densities of $J_D = 1 - 2nA/cm^2$ are achieved by modern CCD processes [44]. The Orbit process used for the FOE chip, however, has not been optimized for dark current, and typical J_D's for it are a good order of magnitude larger than this [13]. To measure this parameter, we merely let the floating gate of the output circuit of the

I/O register sit for an inordinately long time. The gate will fill up with charge and a downward slope is visible in the output. Figure 5-15 illustrates the measured output schematically. First the floating gate is flushed of charge by using the reset voltage to push the charge over the ϕ_4 gate blocking the gate from the output diode. Then the gate is pre-charged and left floating. All of the dark current from the entire output register is assumed to flow into the floating gate at this point, giving rise to the expected slope. The transfer gate to the main array has to be set to block dark current from there from adding in to this signal. All of the levels in the imager are slightly higher than the transfer gate, allowing the dark current developed there to drain through the output diodes at the array of floating gate amplifiers on the other side of the imager instead. Furthermore, the input signal IS and ϕ_4 shield the register from the diodes at either end; this reduces the amount of dark signal lost to the input and output diodes. Thus the collected dark signal is assumed to come only from the I/O register. At a nominal temperature of 30°C measured by a thermocouple at the package, a slope of 25V/sec was measured from the output register, which occupies an area of 780μm $\times$34μm. This results in a J_D of:

$$J_D = \frac{(25V/sec)(1.602 \times 10^{-19}C/e-)}{(2\mu V/e-)(2.65 \times 10^{-4}cm)} = 7.6nA/cm^2 \tag{5.10}$$

As expected, this is quite a bit higher than an optimized process would allow. The dark current levels fluctuated tremendously due to lab temperature variations, and led to the design of the simple cooling system described in Section 4.3.4. In practice, the engagement of the cooling system lowered the background dark level of the images typically by a factor of 2.

5.1.2.4 Optical Responsivity

The last basic CCD parameter of interest is the quantum efficiency of the imager. In order to measure this parameter, the setup shown in Figure 5-16 was used. A programmable voltage source was used to drive a bank of voltage to current converters. This current in turn biased an array of red AlGaAs light emitting diodes (LEDs). These LEDs have a narrow emission peak at 637nm with a width of 40nm. Uniform illumination was obtained by placing an Oriel flashed Opal diffuser in front of the array of LEDs. The resulting array and diffuser were fashioned to fit snugly in the 2 inch aperture in the centering mount of the proximal interface. Overall brightness measurements versus the input voltage (and hence the brightness of the LEDs) were taken at an integration time of 1msec and then the chip was swapped with a calibrated photodiode (Newport 1815 power meter with 818-SL detector) which measured the incident light power under the same conditions.

A plot of the output voltage versus incident light power as measured by the photodiode is shown in Figure 5-17. The linearity of the relationship was excellent, coming in under 0.5% which is comparable to the nonlinearity expected from the cascade of floating gate amplifiers and source followers alone.

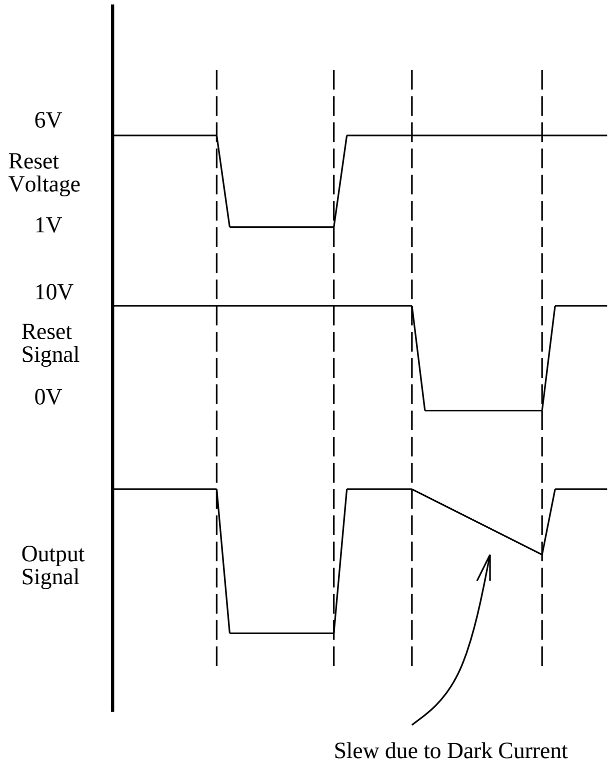


Figure 5-15: Illustration of the measurement of dark current.

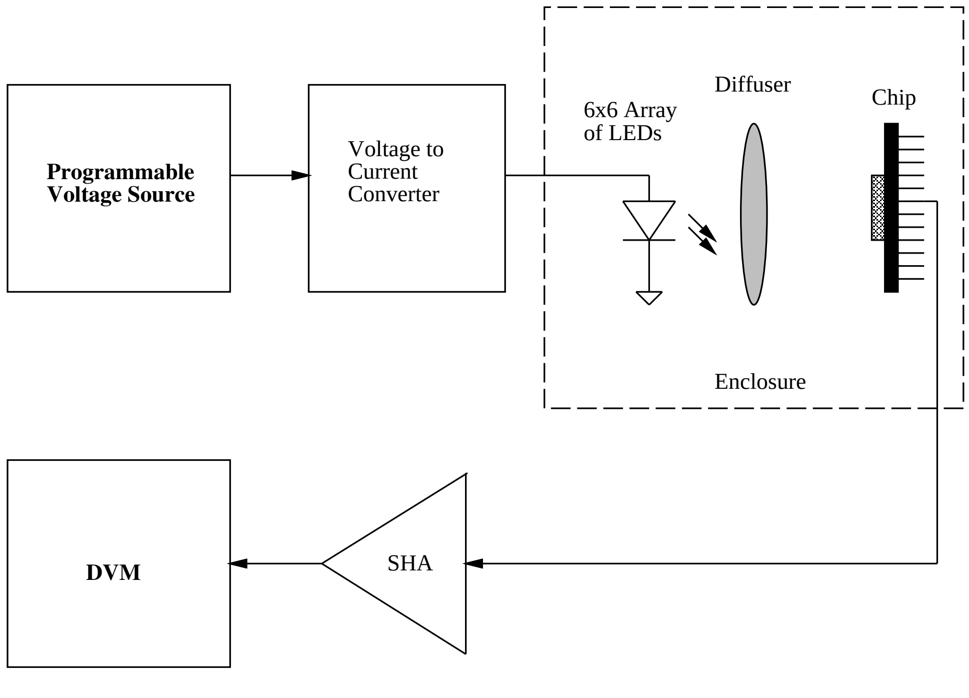


Figure 5-16: Setup for measuring the quantum efficiency of the imager on the FOE chip.

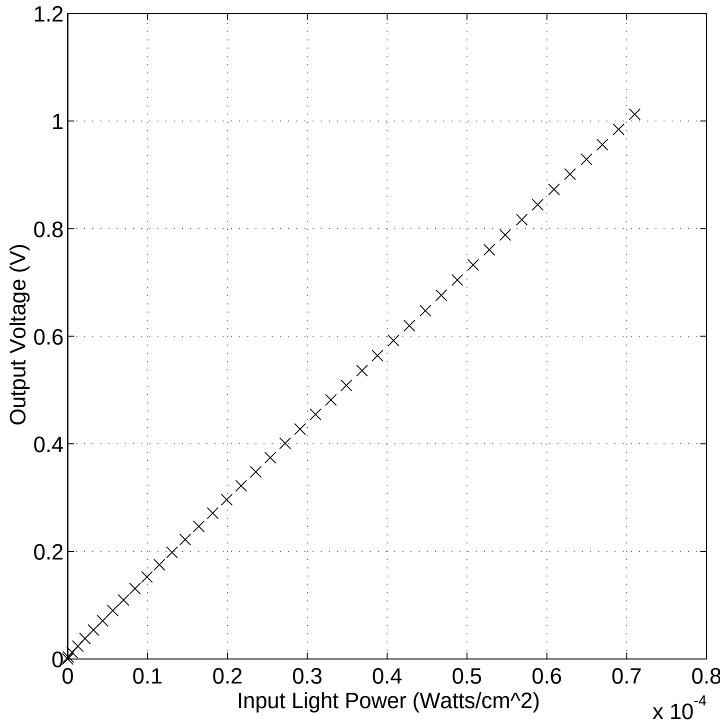


FIGURE 5-17: Plot of the measured output signal as a function of incident light power.

Using this information, we can find the quantum efficiency of our sensor. From the graph, a 1V output signal with a 1 msec integration time corresponds to 0.7e-4 W/cm^2. The optically exposed area in the sensor is 15μm $\times 50\mu$m $= 7.5 \times 10^{-6}$cm^2, and using a sensitivity of 2μV/e- with a source follower gain of 0.97, we find the responsivity to be:

$$R = \frac{(1V)(1.602 \times 10^{-19} C/e-)}{(2\mu V/e-)(0.97)(7.5 \times 10^{-6} cm^2)(7 \times 10^{-5} W/cm^2)(1msec)} = 0.1573 A/W \qquad (5.11)$$

For light at 637nm, 100% quantum efficiency corresponds to $0.52A/W$ (see Figure 3-14), which yields a quantum efficiency of 30% for our sensor. 30% is cited as within the typical range for CCD imagers with front-face illumination [38].

Table 5-2 summarizes the four CCD parameters of interest measured from the FOE chip, all of which are adequate for proper operation.

5.1.3 Electrical Input and Electrical Output

With basic functionality confirmed from our measurements of the test cell and the I/O shift registers, we can now turn to the main array. Using the input structure in the I/O shift register, we can introduce test patterns into the imaging array. In combination with the masking register, this allows probing of the responses of individual processors in the actual array. The split-gate inputs to the array are fairly nonlinear; in practice we can pre-distort our input in software to

I/O FGA Output sensitivity	2μV/e$^-$
Transfer inefficiency	$\leq 7 \times 10^{-5}$ @ 500 kHz
Dark Current	$\leq$ 10nA/cm^2 @ 30°C
Quantum efficiency	30% @ 637nm

Table 5-2: Summary of CCD performance parameters.

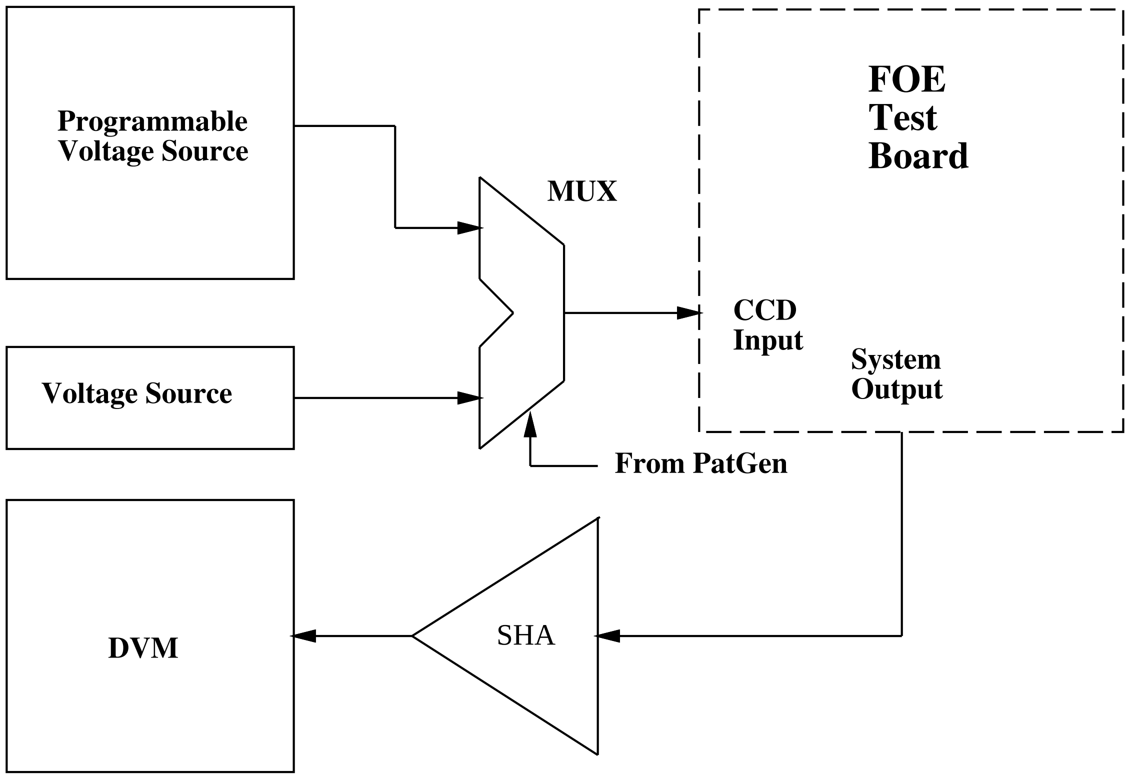


FIGURE 5-18: Setup for introducing test patterns into the FOE chip.

compensate for this effect. The basic test setup is shown in Figure 5-18.

The basic idea is to ping-pong between two voltage levels appropriately to effectively drive each image gradient separately. If we change the input every column, this excites E_t. If we ping-pong the input every row, this drives E_y and if we trigger on every pair of columns, then we drive E_x. Thus, we can verify the functionality of each processor in the main array individually using the masking register. Of course, the gains of the output circuits had to be increased substantially to get sufficient resolution in the outputs. This merely involved using higher valued resistors. Compensation of the output circuits was now more difficult and the gains ended up being set by what compensation could be achieved in practice.

Figure 5-19 shows the output of the absolute value channel under E_t excitation for four different processors spaced along the array. In contrast with the characteristic measured from the test cell, the processors in the actual array present about 10% offsets. Since we can cancel

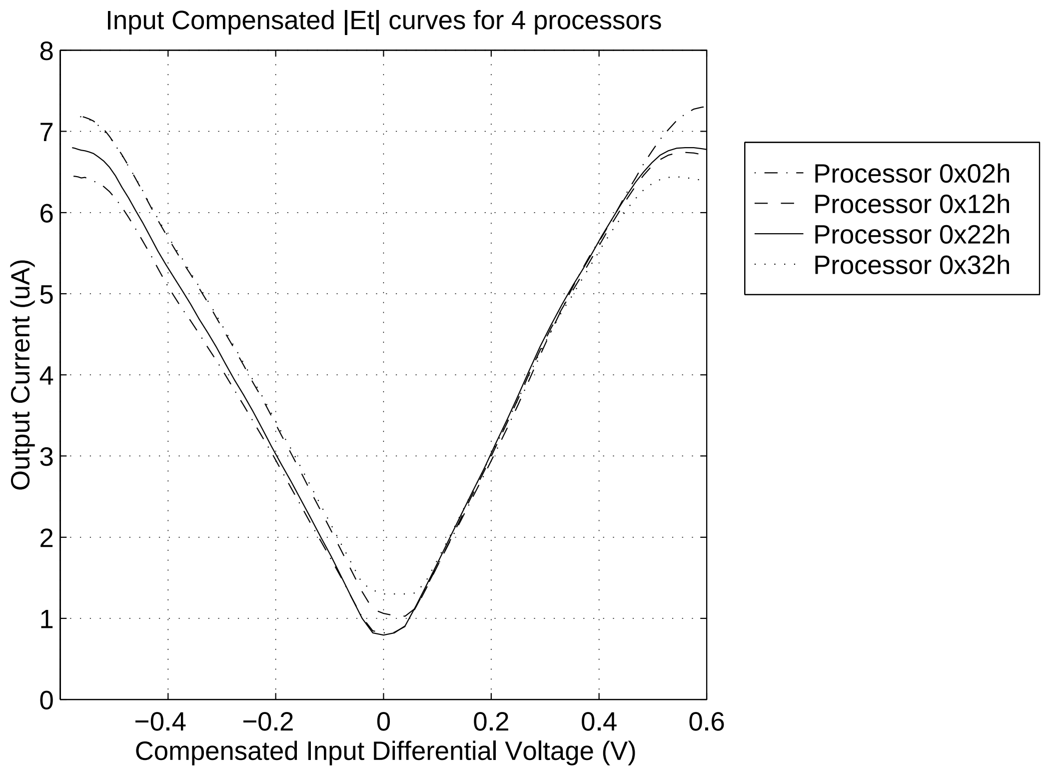


FIGURE 5-19: Measured absolute value characteristic for four processors in the main array under E_t excitation.

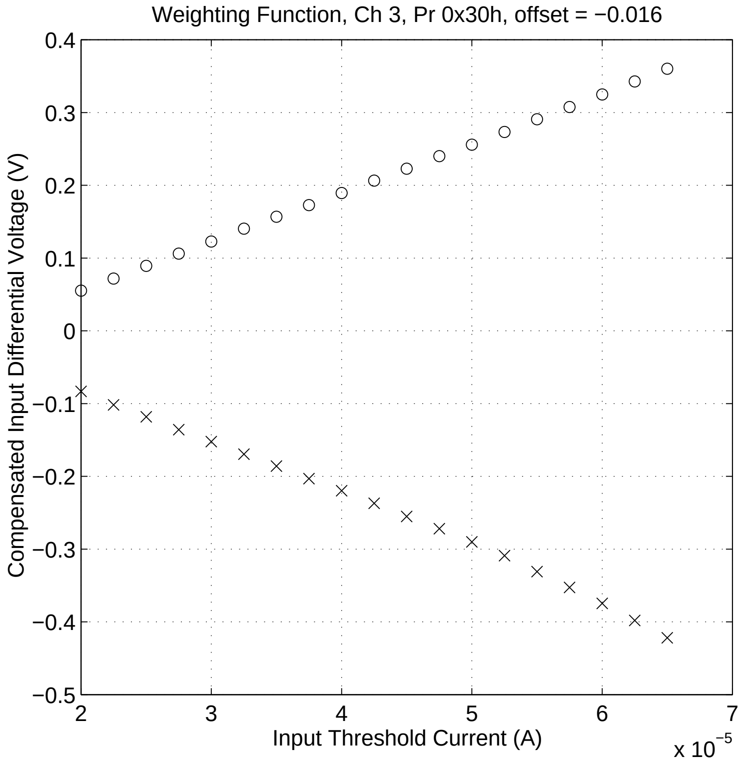


FIGURE 5-20: Weighting function characteristic measured from a processor in the main array.

these offsets, this is not a big problem. The absolute value current gets used by the weighting function however. Figure 5-20 demonstrates a similar 50% weighting plot to the one that we found for the test cell. Deciding whether the processor responded or not was accomplished by measuring the common-mode output current of one of the error channels. Clearly, an offset in the absolute value channel merely changes the effective threshold in the algorithm. Much more distressing is the wide variation in offsets measured for the absolute value circuits between processors. From these four processors alone, a 5% variation in offset is observed. This is sufficient variation to impact performance substantially as this is larger than the typical weighting function width needed by the algorithm. All processors are expected to make the weighting decision in parallel, and no offset cancelation was provided for the cutoff weighting function. Subsequent versions of the chip must either reduce these offsets through careful redesign and/or layout, or provide a means for offset cancelation on-chip. To continue with the current work, the FOE system was reprogrammed to measure via the masking register all of the data, including the offsets, for each full image pair in real-time. The offset cancelation and cutoff weighting function were then performed in software on the test board.

Reconfiguring the test setup for E_y excitation, the family of curves from the y-channel of a processor in the array was measured and the data is shown in Figure 5-21. For the x-channel measurements, the board was reconfigured for E_x excitation. The position encoder was held in a reset state, corresponding to an input position near the bottom of the range. Figure 5-22 shows the resulting family of curves; x_0 was not allowed to go substantially below the x setting. Offsets on the order of 2% are observed for both channels, however this is not nearly as critical as in the absolute value channel.

5.2 Imaging Results: Raw Image Data

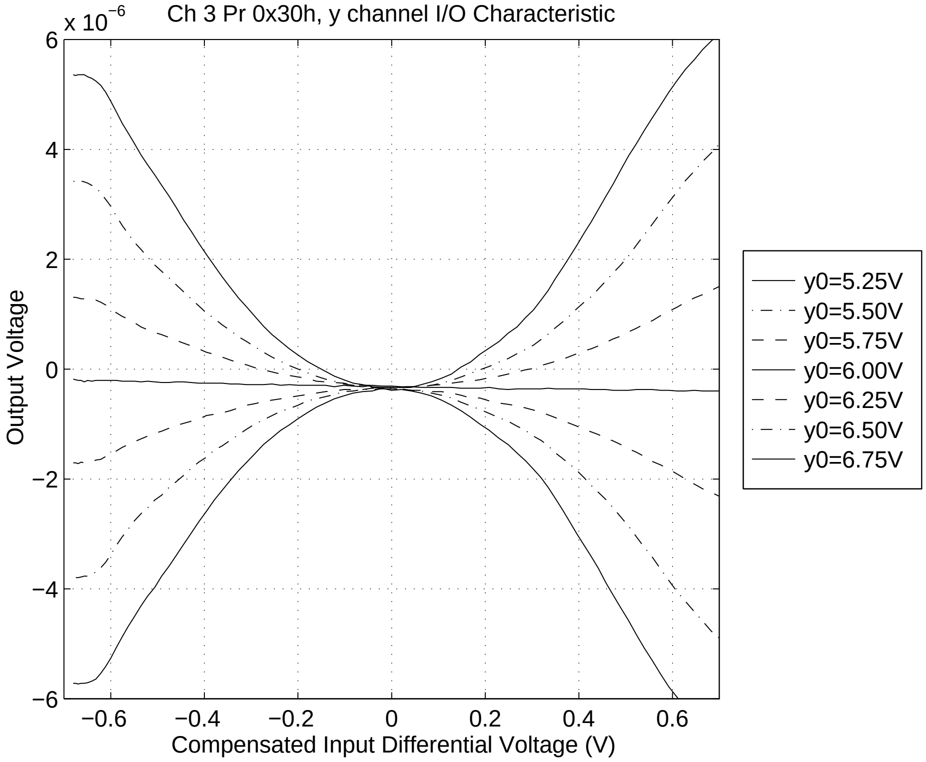
5.2.1 The Target Image

The next stage of the testing process is to perform translational motion experiments, acquire the resulting raw image data, and then simulate the algorithm on the host computer using this data. In the experiments performed with synthetic image data in [23], a single plane with a sinusoidal texture was used. We again adopt this approach due to its simplicity, and a textured plane is placed perpendicular to the carriage motion. When we look along the direction of motion, depth is everywhere the same, and the projection is now orthographic. However, by varying the orientation of the distal mount, not only do we change the location of the FOE, but also the depth of the scene.

Although a sinusoidal texture is useful in simulation because its stationary points actually form continuous contours, creating a sinusoidal texture is not physically practical in the lab. Furthermore, in a sinusoidal pattern the image gradient is very smooth, leading to small signal magnitudes for the system to work with. Hence, we would instead like a Mondrian texture [25] in order to maximize contrast, and hence gradient magnitude. The features required of the texture to give good performance with the algorithm are that it gives rise to a sufficient number of stationary points uniformly distributed about the FOE and whose image gradients are distributed in all directions.

Pathological situations where the image gradient is constrained can, of course, be constructed and in these situations the algorithm will fail. For example, a picture of horizontal bars gives no information about motion in the x direction. Similar problems arise when using vertical bars or checker board patterns. Instead, we decided to use an image consisting of a simple matrix of black disks, arranged in a hexagonal pattern. The image gradient for such a texture is well distributed in all directions. Now, instead of contours as in the sinusoidal case, the stationary points are discrete.

Figure 5-23 shows an example of the resulting stationary points for a single disk. The image gradient is zero everywhere inside and outside the disk. On the perimeter of the disk, the gradient is pointing out everywhere, and so as we go around the disk, the gradient goes through all directions. When the image gradient is perpendicular to the vector from the FOE, we have a stationary point. For disks such as this one, stationary points obviously occur in pairs. With a sufficient number of such disks visible in the image, enough information should be present to reliably estimate the FOE.



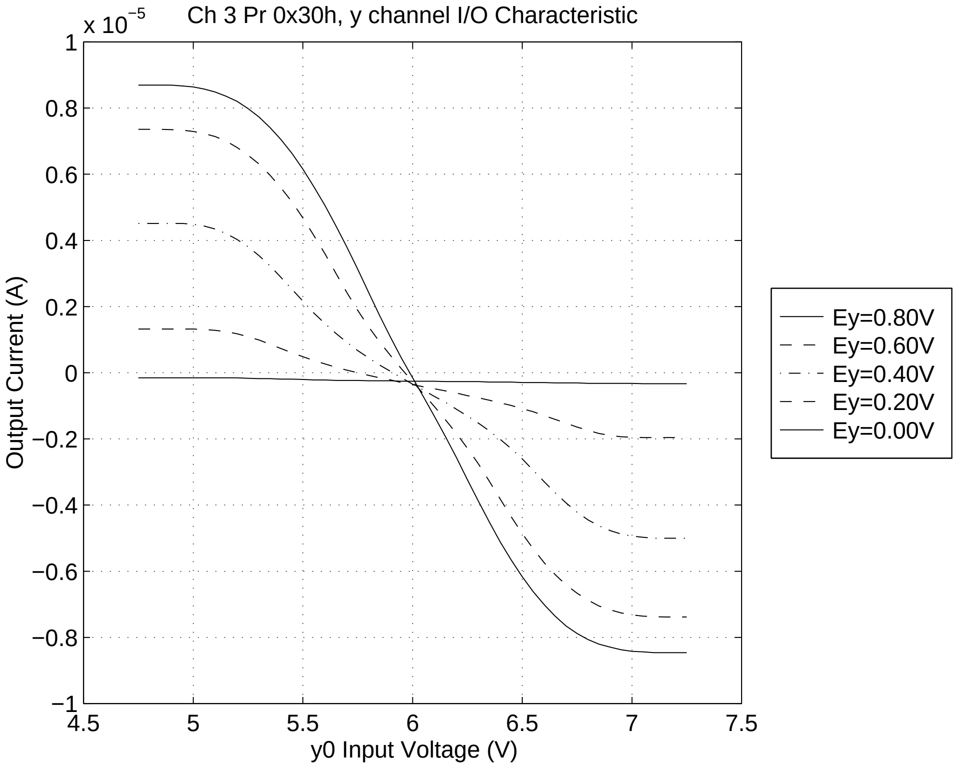
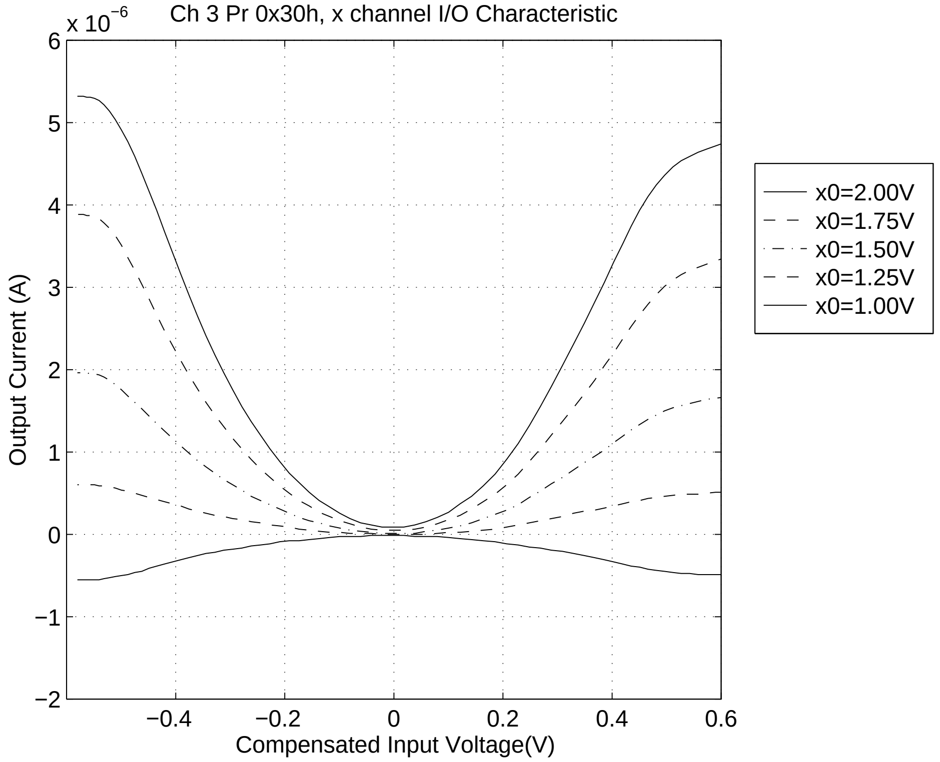


Figure 5-21: Family of curves obtained for the y-channel from a processor in the main array under E_y excitation.



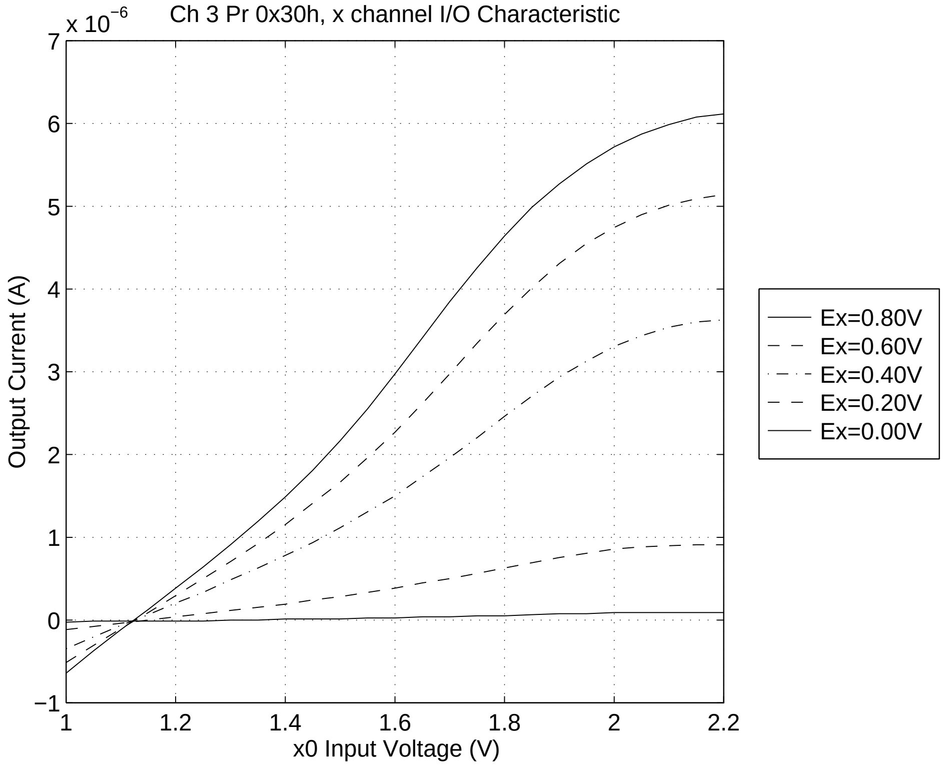


FIGURE 5-22: Family of curves obtained for the x-channel from the bottom processor in the main array under E_x excitation.

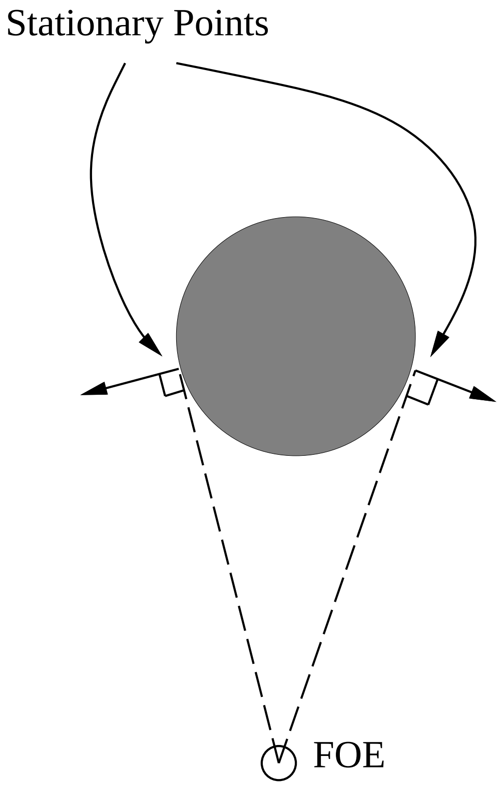


Figure 5-23: Illustration of target scene used in the lab.

5.2.2 Optical/Mechanical Preliminaries

Now we can begin motion experiments. A sample position and velocity profile of the motor system is shown in Figure 5-24. The velocity measurement is done using finite differencing on the position data. The motor control system is optimized for positioning accuracy, and hence the velocity is only constant to within 5%, which is typical for a good positioning system. The velocity may actually be substantially better than this measurement would suggest; the trace feature of the controller uses interrupts to interrogate the servo controller chip performing the positioning. The resulting jitter in the measurement leads to beating when finite differencing is used to look at velocity.

The major drawback of the image carrier becomes obvious when it is included in the system. Not only is there a loss of about 50% from the cable, but there is also substantial loss from the lens system in the cable tip. The lens in the distal end is very small; the aperture appears to be on the order of $1-1.2$mm. As a comparison, a standard 16mm lens with $f/2$ has an aperture of 8mm. Light gathering power goes as the physical area of the lens [77], and hence a reduction by a factor of 8 in lens diameter results in a reduction in light by $8^2 = 64$. Combined with the loss through the cable, the cable/lens system loses about two orders of magnitude of brightness. Although the imager was tested at acquisition times of 1-2 msec, and the settling times of the various components of the FOE chip are adequate to allow operation at these speeds, the

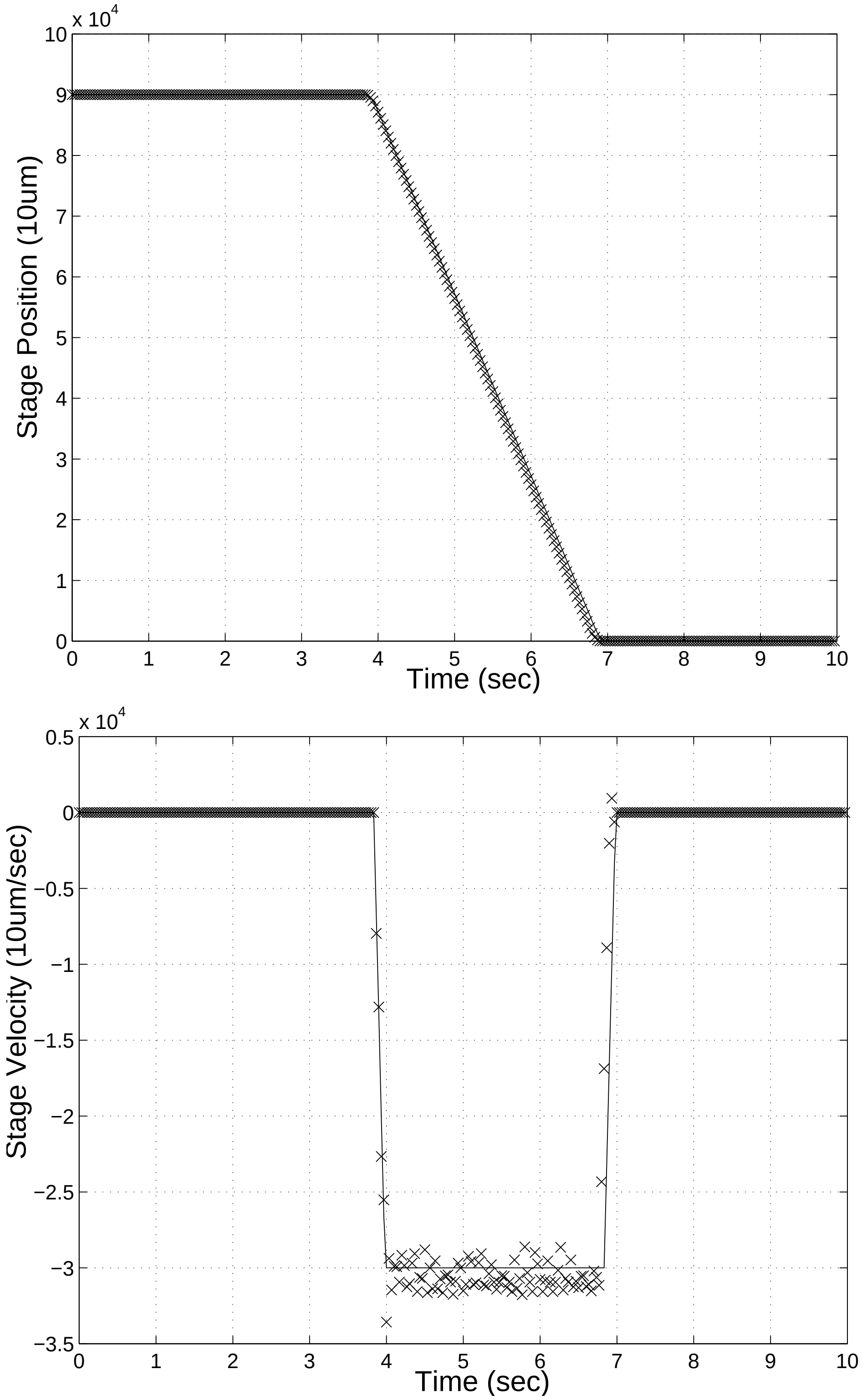


FIGURE 5-24: Position and velocity profile of motor stage.

resulting light input to the chip from the cable is insufficient. To compensate for this loss of light, the frame rate of the system was reduced to standard video rate (30 frames/sec) to get enough signal to use in practice. This reduction of frame rate also allowed for implementation of the cutoff weighting function in software on the test board while still allowing for real-time estimation of the FOE using the rest of the chip architecture.

A sample image sequence is shown in Figure 5-25. Focus is achieved in practice both by looking at image data in real time using a simple X interface, but also by maximizing the edges visible in the output image raster on the oscilloscope. The angle φ was varied until the pattern was aligned with the imager and the position of the optic axis was adjusted until the pattern appeared centered in the image frame. With the cooling system operating, typical contrast ratios were $8 - 10$. The orientation of the distal mount was $\theta = -9°$ and $\phi = 13.5°$, i.e. the camera was pointed left and up. This would seem to position the FOE in the lower right-hand corner. However, the lens system performs an origin reflection resulting in the placement of the FOE in the upper left hand corner as indicated by the cross.

Obviously, in order to interpret the performance of the algorithm we need to know where the FOE actually is. The problem that has arisen then becomes, given the orientation of the viewing direction relative to the motion, where is the FOE? The mapping from the 3-D motion to the FOE location requires knowledge of the location of the intersection of the optic axis with the image plane; this gives offset type information. Additionally, the principal distance is required, and this gives gain type information. The unknown rotation φ should also be estimated, as this rotates the FOE about the optic axis. Finally, the images show a fair amount of lens distortion from the wide angle lens system in the distal tip. This distortion strongly affects the solution from the algorithm when the FOE is near the image boundary, and its parameters should be estimated as well.

5.2.3 Camera Calibration using Rotation

The method we use for estimating the camera parameters is detailed in [78] and involves performing calibration from image data only; no actual measurements of object distances or sizes in the real world need be done. If we rotate a camera about an axis $\Omega = (\Omega_x, \Omega_y, \Omega_z)$ by an angle $-\theta$, then the world point $\mathbf{R}$ will have rotated to a point $\mathbf{R}'$ given by Rodriguez's formula [78]:

$$\mathbf{R}' = \Theta\mathbf{R} = \left[\cos\theta\mathbf{I} + \sin\theta\mathbf{Q} + (1-\cos\theta)\mathbf{\Omega}\mathbf{\Omega}^T\right]\mathbf{R} \tag{5.12}$$

$$\mathbf{Q} = \begin{bmatrix} 0 & -\Omega_z & \Omega_y \\ \Omega_z & 0 & -\Omega_x \\ -\Omega_y & \Omega_z & 0 \end{bmatrix} \tag{5.13}$$

By projection, we find that the image point $\mathbf{r}'$ corresponding to the new world point $\mathbf{R}'$ is

$$\mathbf{r}' = f\frac{\mathbf{R}'}{(\hat{\mathbf{z}}')^T\mathbf{R}'} = f\frac{\Theta\mathbf{R}}{(\mathbf{\Theta}\hat{\mathbf{z}}^T)\mathbf{R}} \tag{5.14}$$

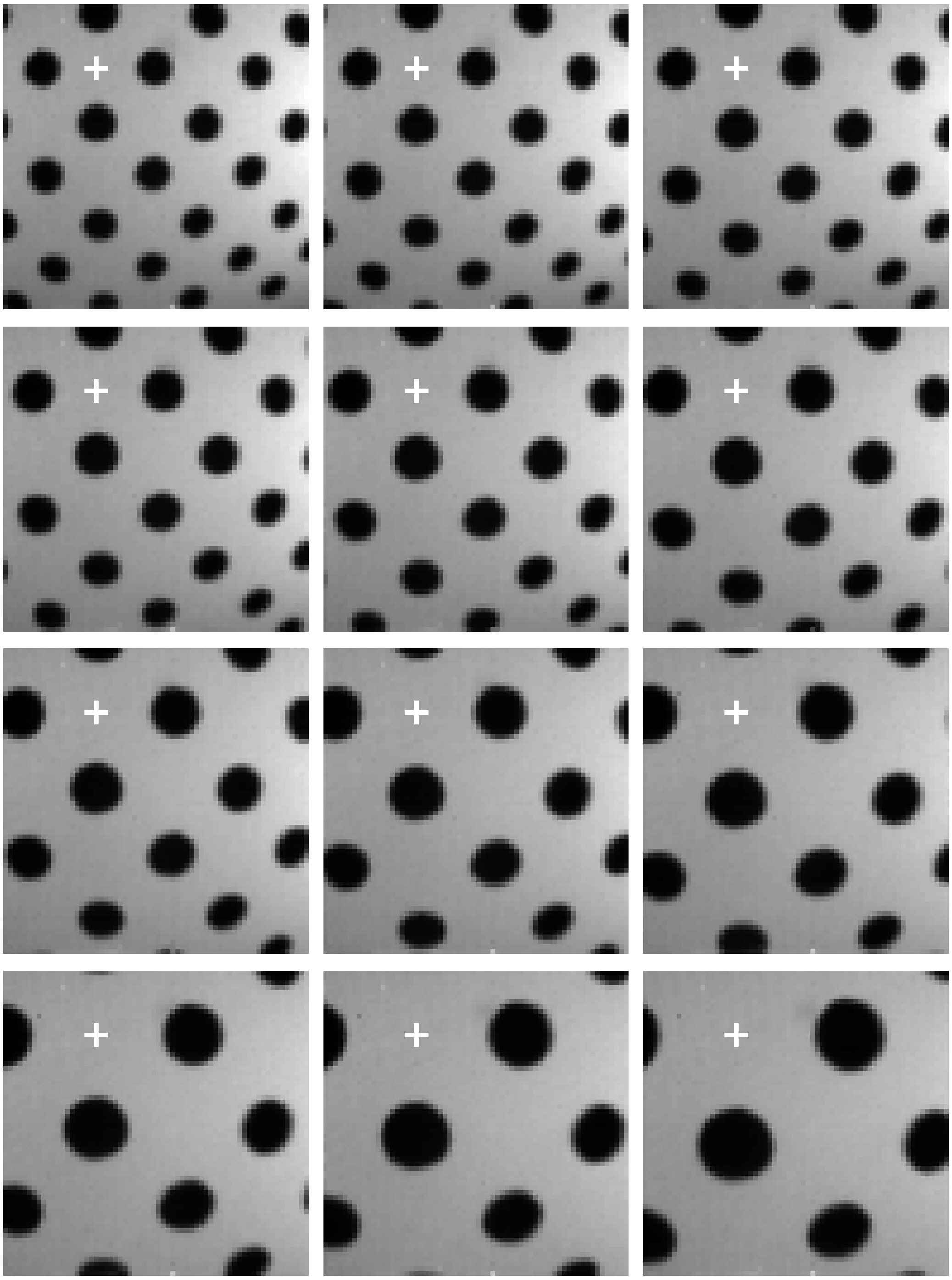


FIGURE 5-25: Sample motion sequence. The settings of the distal mount are $\theta = -9°$ and $\phi = 13.5°$ placing the FOE in the upper left hand corner as indicated by the cross.

The original world point also satisfies the projection equation, and hence $\mathbf{R} = (\mathbf{R} \cdot \hat{\mathbf{z}})\, \mathbf{r}/f$ and so we find:

$$\mathbf{r}' = f \frac{\Theta \mathbf{r}}{(\Theta \hat{\mathbf{z}}^T)\mathbf{r}} \tag{5.15}$$

Notice that the dependence on world points has dropped out of the equation entirely. From this, we can conclude that under pure rotation, the new image point $\mathbf{r}'$ depends only on the camera parameters and the location of the point in the image $\mathbf{r}$ before the rotation. This is the fact on the which the calibration procedure is based. Given the correspondence between a sufficient number of features in the unrotated and rotated images, we can use this relationship to determine the camera parameters.

The location of the intersection (c_x, c_y) of the optic axis with the image plane is unknown and hence points $\mathbf{r}$ in the image plane are described by:

$$\begin{aligned} x_d &= x + c_x \\ y_d &= y + c_y \end{aligned} \tag{5.16}$$

where x_d and y_d are the observed image coordinates. Wide angle lenses typically have large amounts of radial distortion, and the same is true of the distal lens system on the image carrier. When the FOE is near the edge of the field of view, this distortion substantially affects the location predicted by our least squares algorithm. As we shall see shortly, the reasons for this is that information from the image gradient at stationary points close to the FOE contribute more to the solution than those far away, and image gradients close by are near the edge of the field of view and substantially affected by lens distortion. The standard model for lens distortion maps the observable distorted coordinates (x_d, y_d) to the unobservable undistorted image coordinates (x_u, y_u):

$$\begin{aligned} x_u &= x_d + \delta x \\ y_u &= y_d + \delta y \end{aligned} \tag{5.17}$$

Lens distortion is modeled as being typically radial and even. Defining $x'_d = x_d - c_x$ and $y'_d = y_d - c_y$, an even power series in the radius $r'^2_d = x'^2_d + y'^2_d$ is formed:

$$\begin{aligned} \delta x &= x_d \left(K_1 r'^2_d + K_2 r'^4_d + \cdots \right) \\ \delta y &= y_d \left(K_1 r'^2_d + K_2 r'^4_d + \cdots \right) \end{aligned} \tag{5.18}$$

Usually, only the first two terms are deemed significant in the expansion.

Our procedure for calibration is then this. We rotate the camera about each rotation axis in the distal mount individually taking pictures every 5°. From these pictures, we perform feature detection, and feature correspondences between the rotated pictures are found. Given the simple nature of the images used in the system, this is easily done by binarizing the images, grouping the resulting points in the dark regions using distance as the clustering metric, and

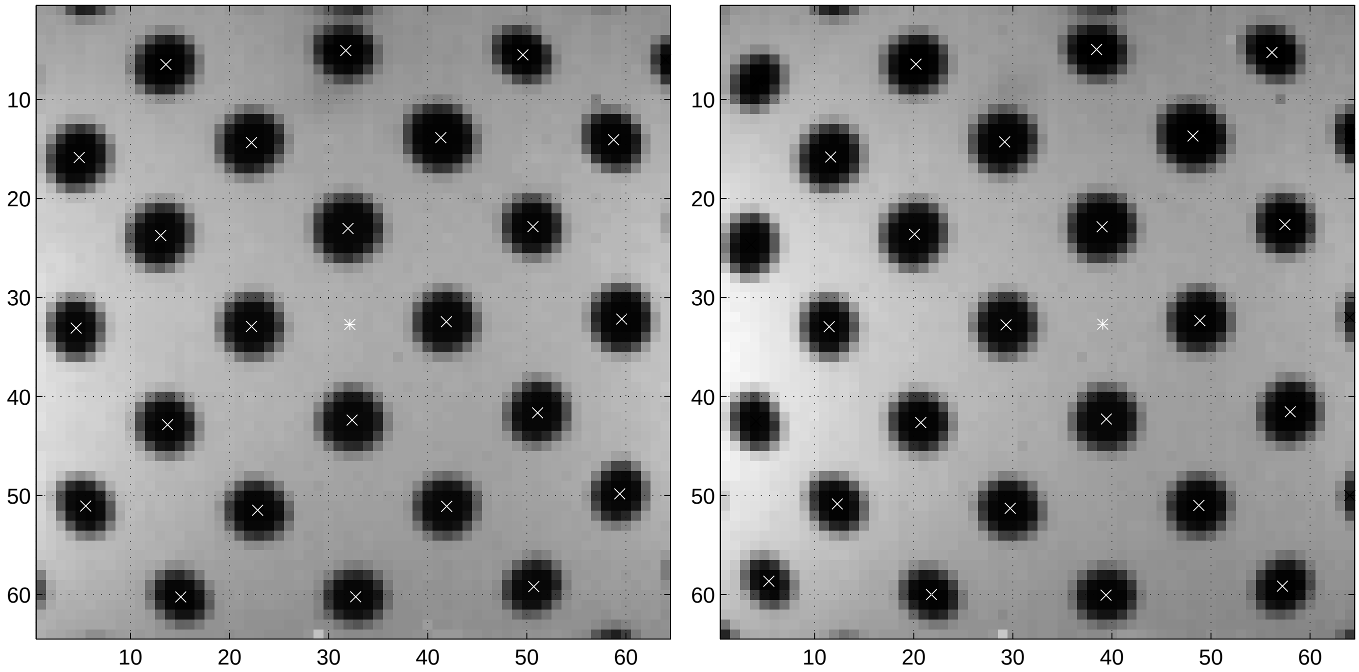


FIGURE 5-26: Target scene with $\theta = 0$ and $\theta = 5°$. Detected features are indicated by "x"s while the center of the image target is indicated by "*".

then performing windowed centroids to fine-tune the feature locations. An example of the results is shown in Figure 5-26. Between image frames, correspondences were found based on minimum distance, as the change in feature locations at each rotation was insufficient to make this an ambiguous operation. The resulting correspondence information is fed to the nonlinear optimization code of [78] which estimates the principle distance f, the location of the optic axis (c_x, c_y) as well as the distortion parameters (K_1, K_2) and the two rotation axes using the LMDIF routine from the MINPACK-1 software package [79]. From the two rotation axes estimated by the program, we can additionally estimate the rotation ϕ in the distal mount. Typical calibration results are shown in Table 5-3. The results of the calibration indicate that the field of view is $\max(\phi_v) = 29.5°$ on the diagonals and $\min(\phi_v) = 21.8°$ along the x and y axes. The cable has a field of view of $\phi_v = 40°$, and using this we find that the imager sees only 29% of the output spot. This corresponds to an oversampling ratio of ≈ 2.8. This is the result of the extra magnification and is a little less than half the ideal of 6.2. Addition of an aperture in the distal lens system reduced the remaining fixed pattern noise substantially, and reduced the overall brightness as well.

5.2.4 Imaging Results: Finding the FOE

Using the parameters given by the camera calibration, we can predict the location of the FOE during a motion transient. To test the algorithm performance using the raw image data and compare the results with the predicted location, the FOE was strobed over the image plane

Imaging Parameter	Calibrated Value
f	79.86 pixels
c_x	31.33 pixels
c_y	33.39 pixels
K_1	5.96e-05 / pixel^2
K_2	1.03e-08 / pixel^4
ϕ	0.86°

Table 5-3: Summary of FOE chip camera calibration parameters.

and the resulting image data from the motion transients stored for processing. Each experiment involved setting the distal mount angles (θ, ϕ) appropriately, and running the stage back and forth while 128 image pairs were captured on the FOE board. Due to the reduced frame rate, the travel of the stage was not sufficient for 128 frame pairs. In this case, image acquisition was synced with the motion which was repeated until 128 frame pairs of data were captured. Time periods during acceleration and deceleration were also eliminated. The algorithm was then performed on the 128 frame pairs in the host computer, Figure 5-27 shows the gradient map selected by the weighting function overlaid on top of one image in that pair; the FOE is indicated by the x. From the 128 image pairs, a mean location of the FOE, along with a standard deviation over the transient, are computed. The results of these experiments are shown in Figure 5-28.

Neglecting lens distortion, the locations of the FOE as provided by the calibration are shown as o's, while the results of the algorithm are shown as x's. Near the center of the image, the algorithm closely estimates the predicted location. Near the edge of the image, the algorithm deviates substantially from the actual FOE and this deviation acts to move the solution towards the center of the image plane.

5.2.4.1 Biases in the Solution

The are a variety of effects which come into play which result in a deviation of the estimated FOE towards the image center:

- Nonuniform illumination leads to image gradients at non-stationary points that contribute to the solution.
- "Nearby" stationary points influence the solution more than "far away" stationary points because
  - a spatial drop-off in sensitivity to deviations in the output and
  - the range of $|E_t|$ grows with the distance away from the FOE, and as a result the cutoff weighting function selects fewer points.

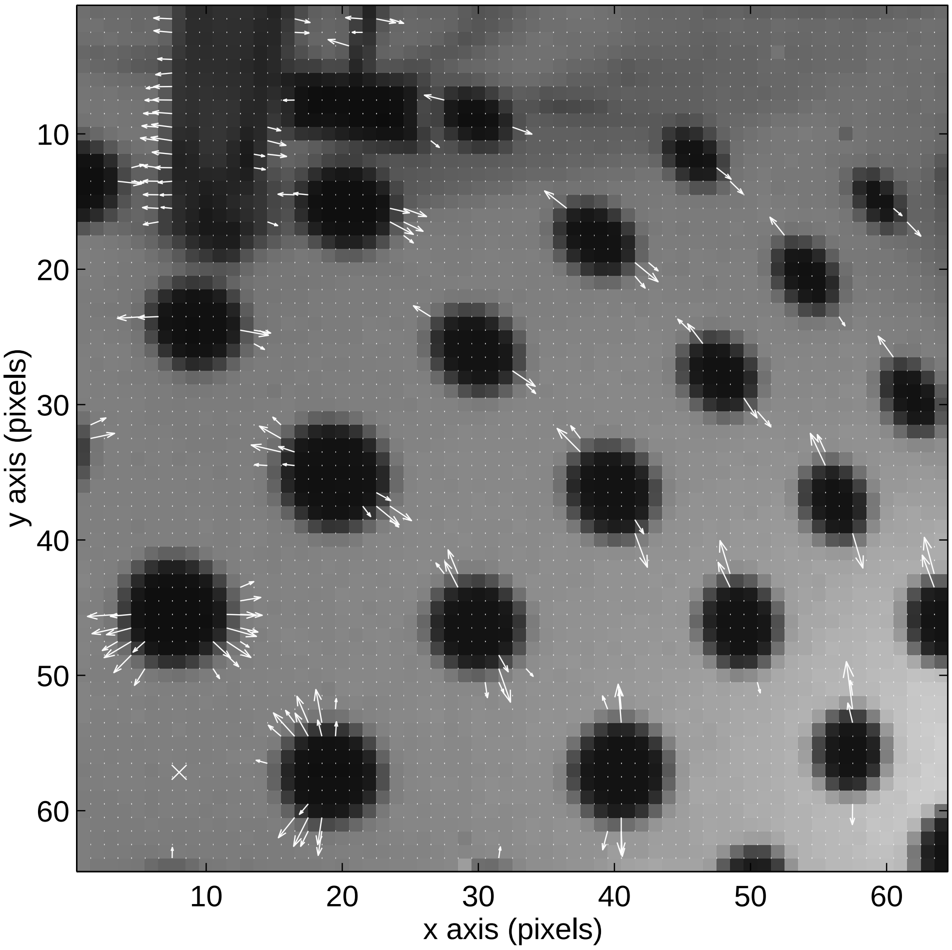


FIGURE 5-27: Example of the algorithm in action. The settings of the distal mount are $\theta = -18°$ and $\phi = -18°$ placing the FOE in the lower left hand corner as indicated by the x. The track of the motor system is visible in the upper left hand corner.

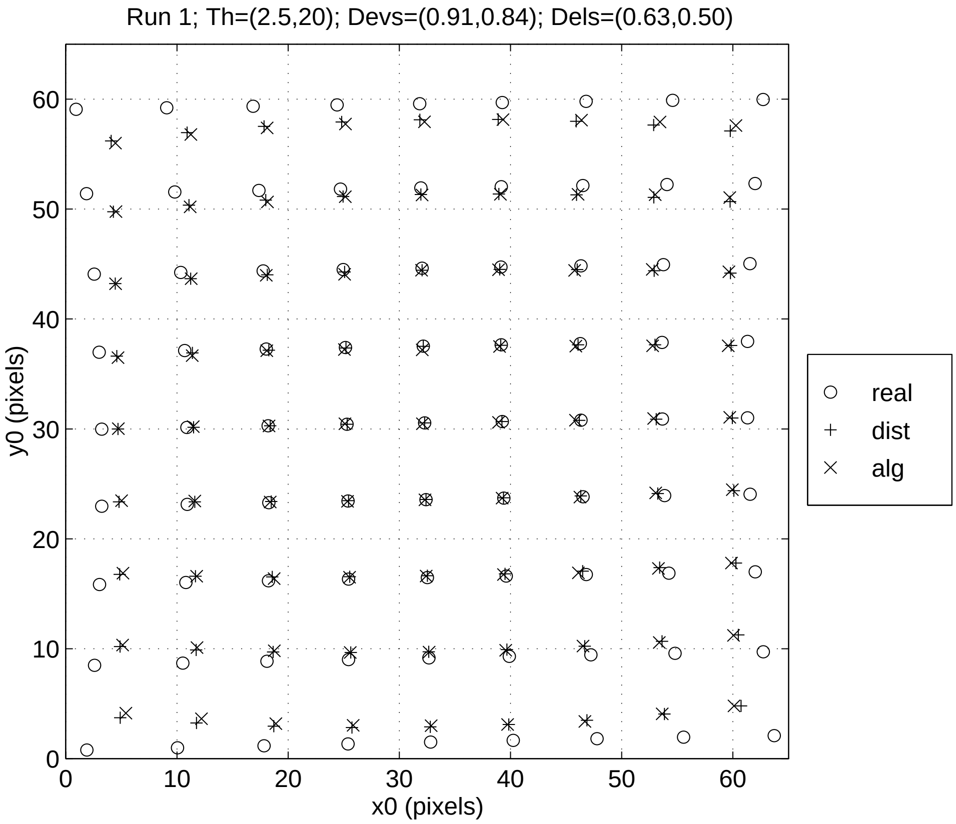


Figure 5-28: Results of raw image data taken from imager on FOE chip. The o's indicate the locations of the FOE as computed using the camera parameters and neglecting distortion. The +'s indicate the distorted FOE locations while the x's indicate the results of the algorithm.

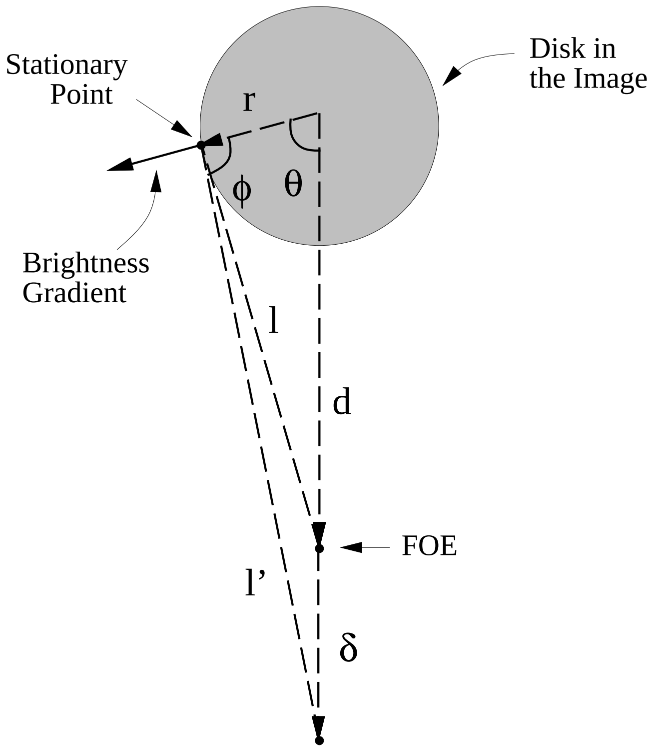


FIGURE 5-29: Geometry used to consider sensitivity of the stationary point weighting to deviations in the solution.

- Lens distortion dominates at the image boundaries.
- Bands around stationary points become attractive.
- Non-uniformity of the distribution of stationary points about the FOE.

The target used in the setup is finite and when the system looks sufficiently off-axis, the target ends up taking only a small portion of the actual image, and the viable stationary points are an even smaller subset. The majority of the image merely shows the smooth drop off in illumination that occurs towards the light sources, which were placed to the left and right of the target. In this image region, the gradient is small but not zero. E_t is also very small, and hence all of this data is selected for contributing towards the solution. Since these points therefore form the overwhelming majority in the computation, the algorithm is strongly biased. When a nonuniform model for lighting was included in synthetic data made to closely match the observed data, this effect was also apparent. To remove this part of the bias, a simple threshold on the image gradient magnitude was used in Figure 5-28. Now, only points where $|E_t| < \eta$ and $|\nabla E| > \xi$ are allowed to contribute to forming the solution. This improves the situation somewhat, but the solution is still biased towards the center when the answer is near the edge.

Consider the simple situation shown in Figure 5-29. Here, the FOE is at a distance d away from the disk as shown in the Figure. The stationary point on the perimeter of the disk is a distance l away. At the stationary point, $p = \nabla E \cdot (\mathbf{r} - \mathbf{r}_0) = 0$ and of course $\Rightarrow \phi = 90°$. If we look at the correction term in the output, we see that:

$$e = \sum W(E_t)\nabla E \nabla E^T(\mathbf{r} - \mathbf{r}_0) = \sum W(E_t) p \nabla E \tag{5.19}$$

Hence, the correction is formed by the sum of the brightness gradients at the stationary points weighted by the dot product p. We are interested in the contribution of a stationary point to the correction if we perturb the solution away from the actual FOE in the y direction to a point $d + \delta$. Using the law of cosines and the fact that $\cos(\theta) = r/d$, we note that:

$$\begin{aligned} l'^2 &= (d+\delta)^2 + r^2 - 2(d+\delta) r \cos(\theta) \\ &= (d+\delta)^2 - \left(1 + \frac{2\delta}{d}\right) r^2 \end{aligned} \tag{5.20}$$

Once again using the law of cosines,

$$(d+\delta)^2 = l'^2 + r^2 - 2l' r \cos(\phi) \tag{5.21}$$

Substituting and simplifying, we find that:

$$\cos(\phi) = -\frac{r\delta}{dl'} \tag{5.22}$$

$$\begin{aligned} p &= \nabla E \cdot (\mathbf{r} - \mathbf{r}_0) \\ &= |\nabla E l' \cos(\phi)| \\ &= |\nabla E \left(\frac{r\delta}{d}\right)| \end{aligned} \tag{5.23}$$

Thus, the weighting given to the gradients in the correction sum due to a perturbation in the output drops inversely with distance. This implies that the contribution to the correction due to "close" stationary points is larger than for "far" stationary points, and as a result the solution is more sensitive to information nearby.

Consider now the situation depicted in Figure 5-30. Here we have simplified by eliminating the y-dimension and we are looking down on the projection when the imager is looking off axis. The depth in this case varies as:

$$Z = Z_o + d \sin\theta \tag{5.24}$$

By projecting onto the image plane, we find:

$$\frac{x}{f} = \frac{d\cos\theta}{Z} \tag{5.25}$$

Using this we can find a simple expression for the image depth depending only on the position x in the image plane:

$$Z = Z_o \left(\frac{1}{1 - \left(\frac{x}{f}\right)\tan\theta}\right) \tag{5.26}$$

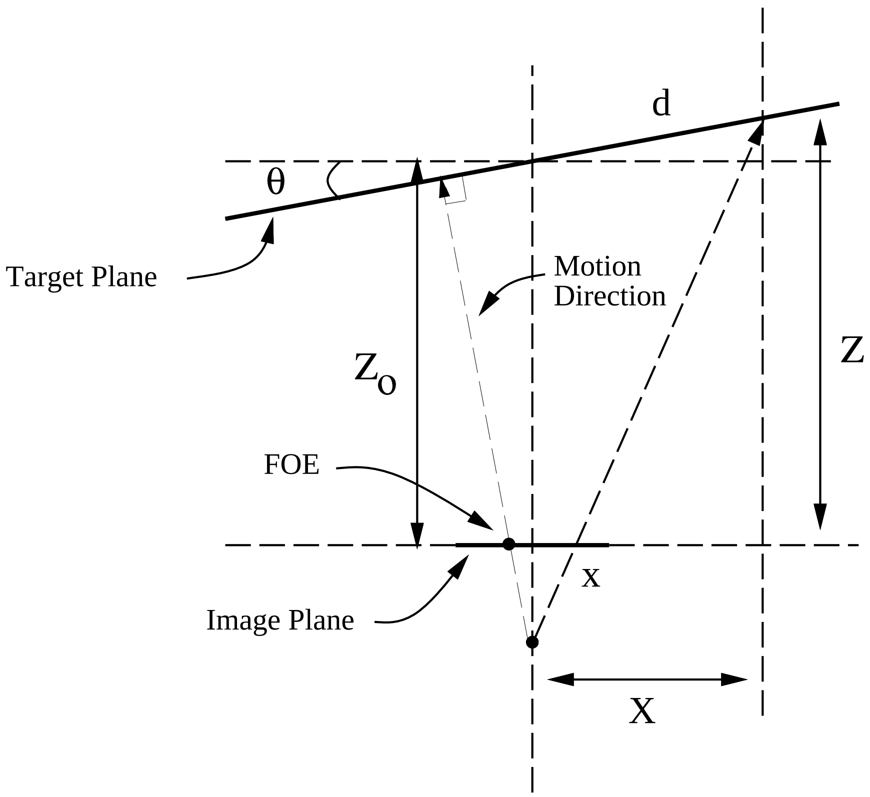


FIGURE 5-30: Geometry for considering the effects of the cutoff weighting function.

Now, we will consider an image gradient that points strictly away from the FOE. Using the brightness change constraint equation, we find:

$$\begin{aligned} |E_t| &= \left|\frac{\nabla E \cdot (\mathbf{r} - \mathbf{r}_0)}{\tau}\right| \\ &= \left(\frac{f|\nabla E|}{\tau_o}\right) \frac{x}{f} \left[1 - \left(\frac{x}{f}\right) \tan\theta\right] \end{aligned} \tag{5.27}$$

Hence, the time derivative E_t grows the further away the feature is from the FOE. Near the FOE, this growth is linear. Further away, the growth is somewhat reduced by the foreshortening due to the perspective, but typically not sufficiently enough to reduce the overall time derivative. Now consider our case with the disks. Disks far away from the FOE in the image are foreshortened by projection. Additionally, the edges of the disk where the gradient points away from the FOE have increasingly larger $|E_t|$ the further out we go. Still, the stationary points exist on the perimeter of the disk. This means that as we traverse the perimeter of the disk, the time derivative E_t goes through zero more quickly the further out from the FOE the disk is. The cutoff weighting function picks up points in a band about $E_t = 0$, and so the number of these points drops off as we look at stationary points further away from the FOE. Thus, the resulting bands used by the algorithm get smaller and therefore contribute less to the computation, leading once again to a decreasing dependence on distance away from the FOE. Given the pixelation of the imager

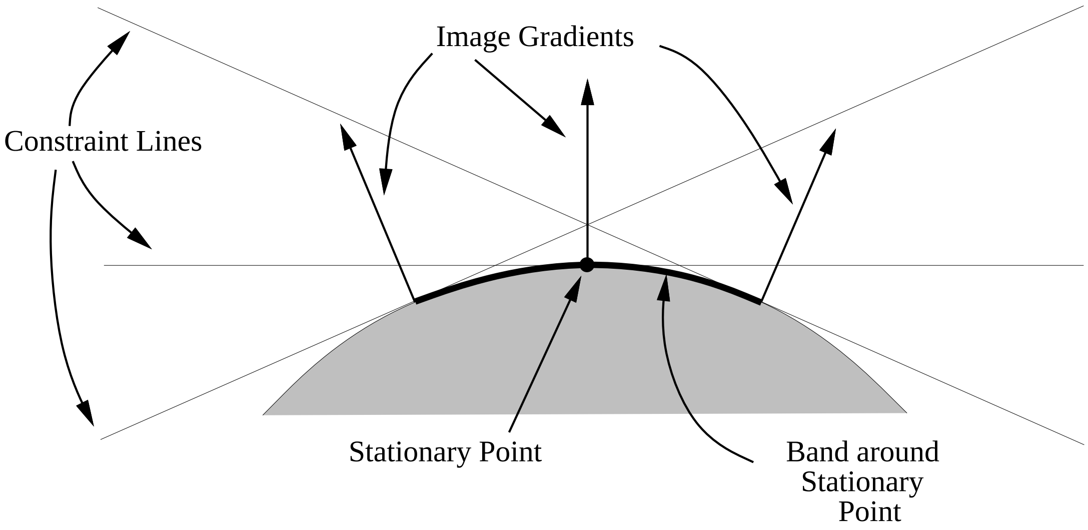


Figure 5-31: Simple geometry of a band about a stationary point.

and the finite differencing used to estimate the brightness gradients, E_t might go through zero so rapidly that no point is identified as a stationary point by the cutoff weighting function.

Due to this spatial drop-off in sensitivity of the output, we can consider stationary points far from the solution as contributing less than those nearby. In effect, the ones far away can be thought of as performing a "coarse" positioning, while the nearby ones do the "fine" positioning. This conceptual picture is especially convenient when the solution is found via a finite-gain feedback loop using the correction. When the estimate is near the right value, the correction forces due to far away stationary points drop off relative to the forces due to the nearby stationary points, which in turn then end up dominating the fine-tuning of the solution. This has special relevance because when the FOE is near the image edge, stationary points nearby are also near the edge and hence strongly affected by lens distortion. If we take the FOE location and use the lens distortion, we can find a new FOE position *as seen through the lens.* These points are indicated in Figure 5-28 by the +'s, and the algorithm closely matches these points near the edge, as opposed to the actual locations of the FOE shown by the o's. Since this distortion is found by calibration and is reversible, when examining accuracy we will consider these distorted locations to be the desired values.

Having produced these new FOE locations, we find that errors in FOE location are now sub-pixel, with the maximum deviation of the mean location of the estimated FOE locations away from the distorted FOE locations under 1% full scale for each component. However, there still seems to be a discernable bias towards the center. Consider the situation shown in Figure 5-31 Here, we examine a single stationary point at the apex of the disk. In the absence of any other information, the gradient at the stationary point only constrains the FOE to lie on the horizontal line shown. On this line, the dot product p is zero. If we include nearby points into the computation, we get a band about the stationary point contributing to the solution,

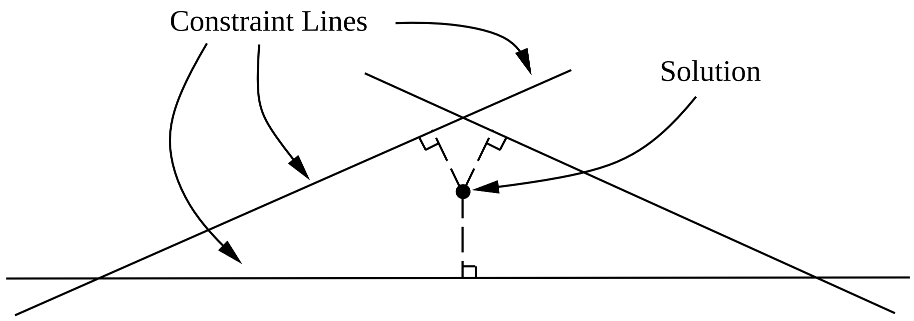


FIGURE 5-32: Least squares solution given by the center and endpoints of a band about a stationary point.

each one of which gives a similar constraint line. In figure 5-32, the constraint lines for the center and the two points at the edge of the band are shown. A standard way for considering least-squares is to attach a spring from the solution to the constraint lines; the least-squares solution minimizes the energy stored in the springs [25]. This is the same as saying the solution minimizes the sum of squares of the perpendicular distances to the constraint lines. Considering only the 3 constraint lines in Figure 5-32, we see that introducing the bands about the stationary point creates an attraction to the band, whereas x_0 should be completely unconstrained by the stationary point. Once we consider each band of stationary points to be weakly attractive, it becomes clear that symmetry of the image also affects the FOE solution, and tends to move it towards the center. When the FOE is placed near the edge of the image, many constraints are missing because the finite extent of the image truncates them away. These constraints would normally act to somewhat counterbalance the rest of the constraints in the image and produce the correct result. However, as shown in Figure 5-33, constraints now are no longer placed uniformly around the FOE, and the unbalancing of the result causes the solution to be drawn towards the constraints in the image, and hence towards the center. This effect is most notable when features with stationary points on them leave the field of view as the camera translates.

5.3 Processed Data Results: Finding the FOE

Finally, we now turn to estimating the FOE in real-time using the FOE chip in a feedback loop. Due to the reduction in frame rate because of the low levels of light available from the image carrier, we can actually raster out all of the data as well as the offsets over an entire image pair via the masking register. The offset cancelation as well as the cutoff weighting function and the new gradient magnitude thresholding can all be done in software and we will still be able to estimate the FOE in real-time. All of this extra processing by the test board can be accomplished by taking advantage of the extra time afforded by slower frame rate.

Figure 5-34 shows the measured offsets from the absolute value and gradient channel. Both of these channels exhibit major offset problems when compared with the measured offsets from

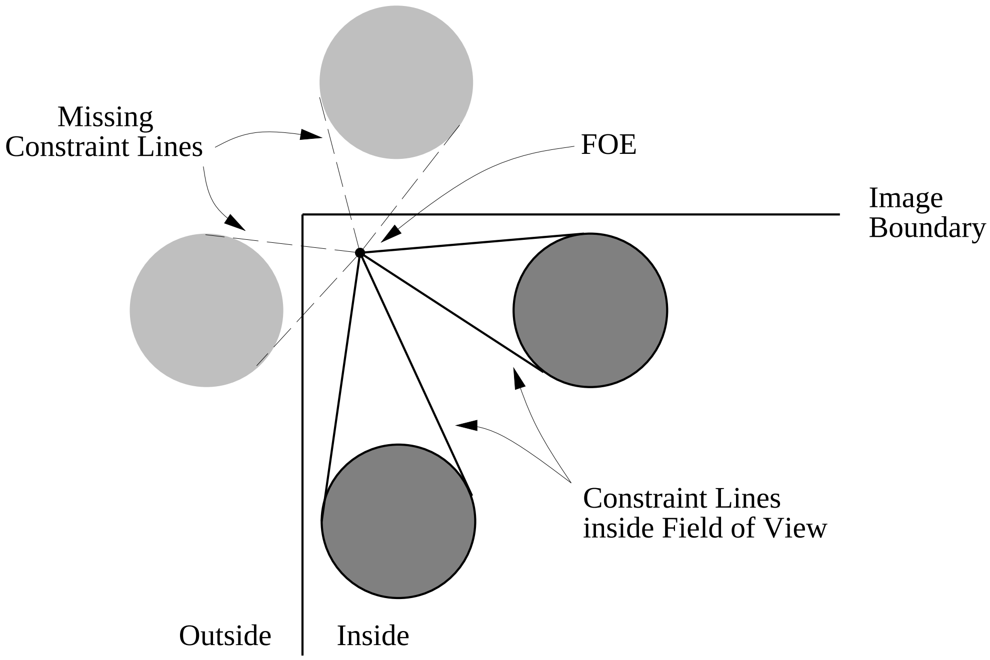


FIGURE 5-33: Illustration of the situation when the FOE is near the image boundary. Constraints from features that would normally counteract the stationary band attraction are absent due to the finite image size.

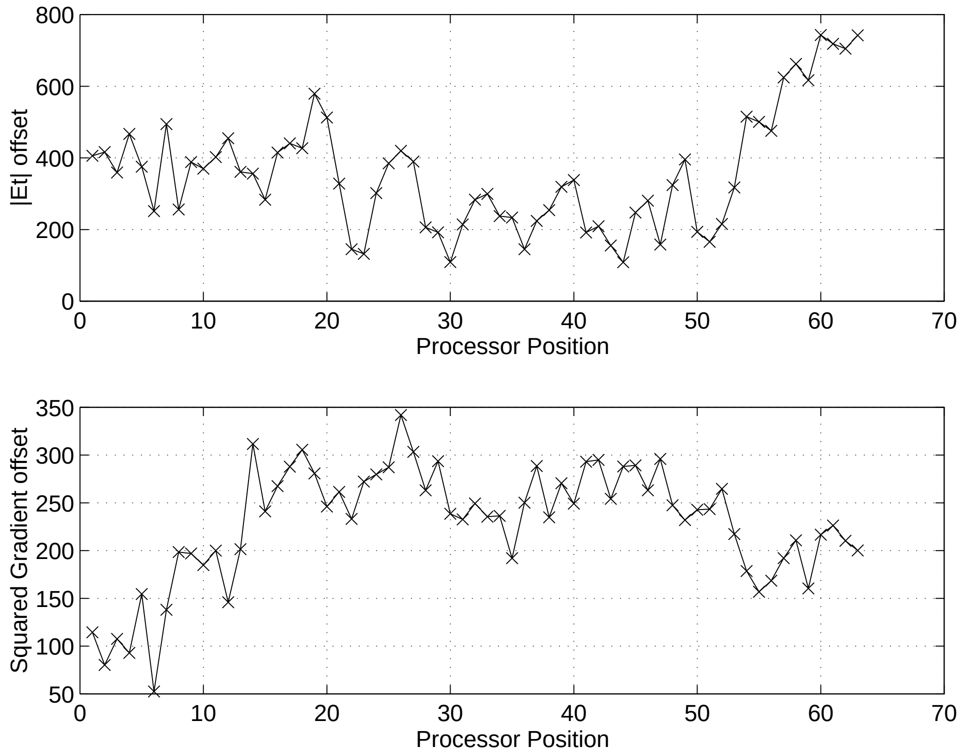


FIGURE 5-34: Measured offsets for the entire column of processors for the absolute value channel and the gradient magnitude channel

.

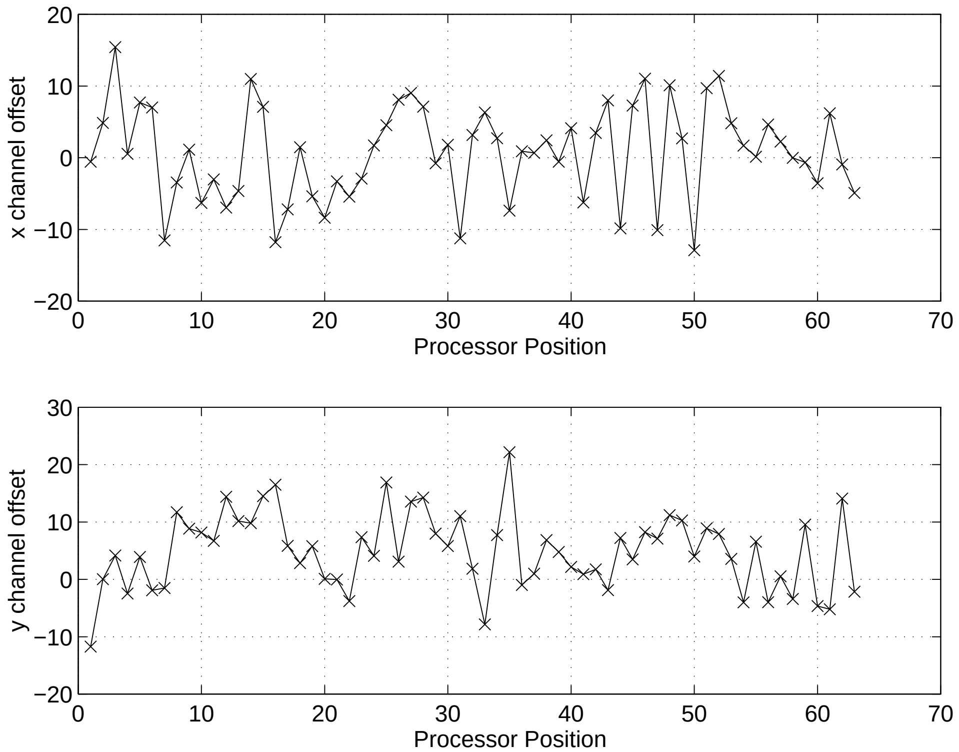


FIGURE 5-35: Measured offsets for the entire column of processors for the absolute value channel and the gradient magnitude channel

.

the x and y error channels, which are shown in Figure 5-35. However, these offsets can be cancelled off-chip and we can still get reasonable performance with the closed loop system. Figure 5-36 shows a mesh plot of the $|E_t|$ channel (after offset cancelation) during a motion transient. The positioning of the distal mount was placed looking straight ahead. Notice that the disks show the expected increase in the magnitude of E_t the farther away they are from the FOE. If we display this instead as an image, we have the picture shown in the left hand side of Figure 5-37. The locations of the stationary points on the disks in the image are clearly visible. The right hand side of the figure shows the binary image that results from application of the cutoff weighting function. From this data, an additional problem is observed in the $|E_t|$ channel. A small feed-through of the input image appears superimposed on the data. This leads to a pedestaling of $|E_t|$ at the disks and a reduction in the minimum achievable sensitivity for the weighting function. This effect was not sufficient, however, to prevent operation.

The reason for the image dependent offset is due to another layout error that was found on the FOE chip. Due to spacing constraints, the outputs of the source followers in the floating gate array were routed over the floating gates themselves. The section of each metal1 wire which ran over the floating gates was 15μm long and 3μm wide. With a metal1 to poly1 parasitic capacitance of 0.042 fF/μm^2 this amounts to about 1.89fF. Since the load capacitance

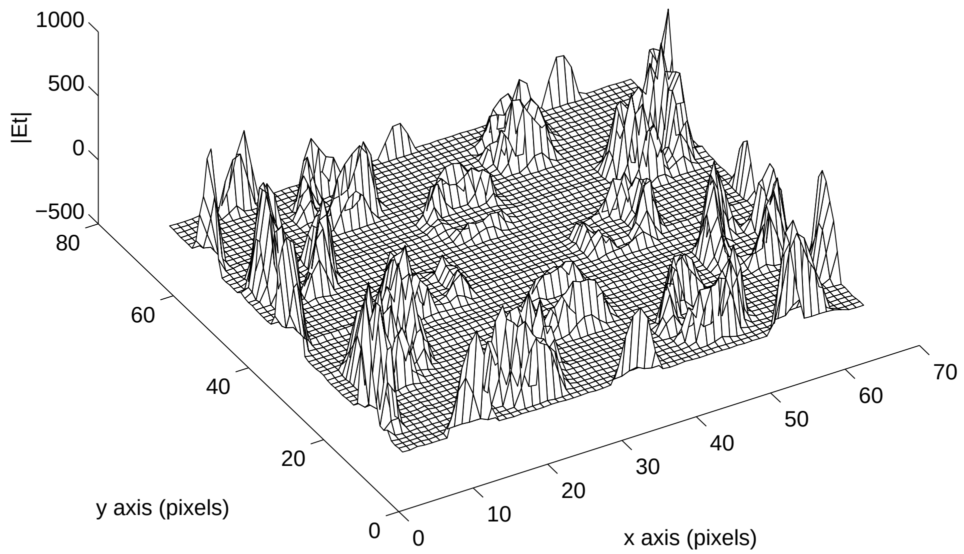


FIGURE 5-36: Mesh plot of $|E_t|$ as calculated by the FOE chip with frontal motion.

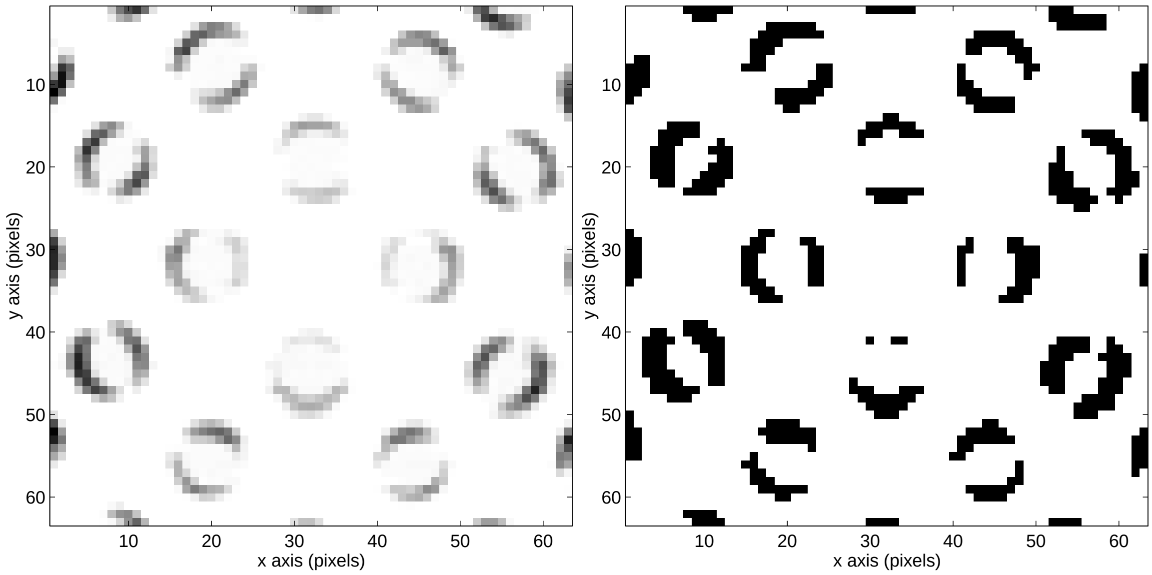


FIGURE 5-37: The image of the $|E_t|$ channel during frontal motion is shown on the left, and on the right is the binary image representing the result of applying the cutoff weighting function.

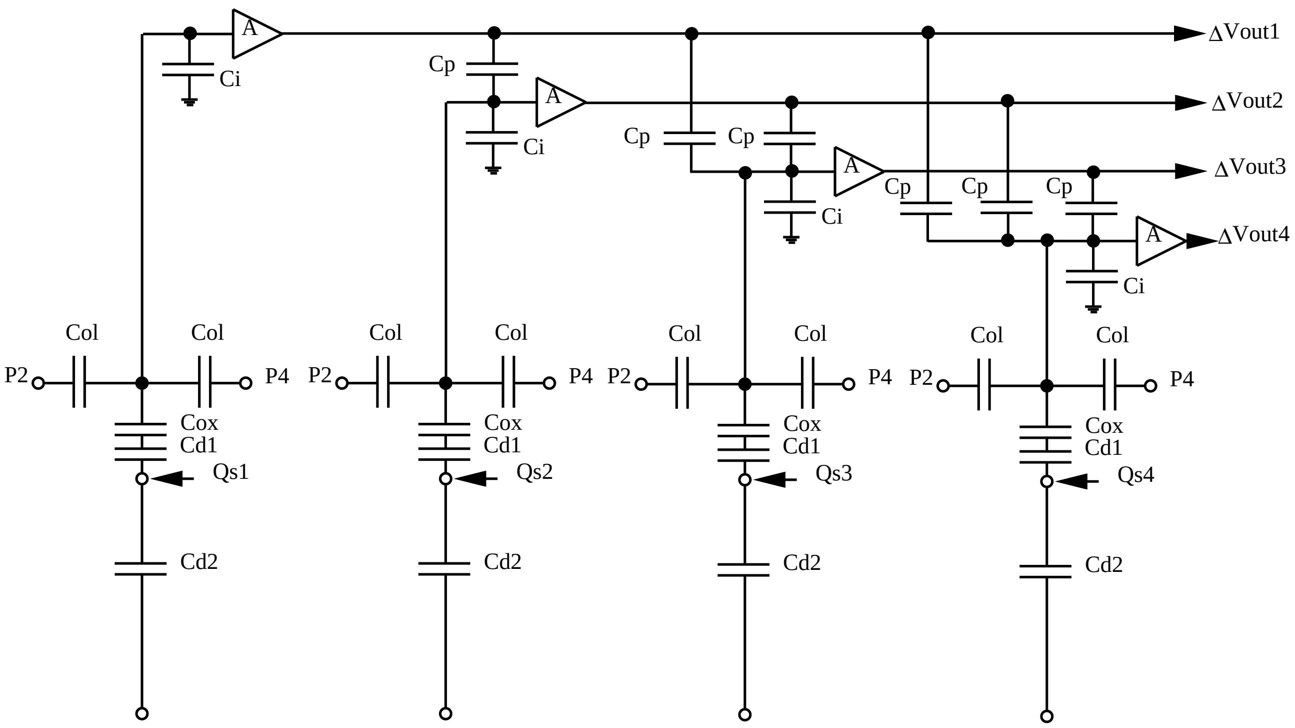


FIGURE 5-38: Lumped circuit model for the set of four floating gate amplifiers including the effects of the layout error.

in the floating gate amplifiers is only 60fF, this is about 3% of the load. The overall lumped circuit model for the set of four amplifiers is shown in Figure 5-38. These parasitic capacitances lead not only to changes in the effective sensing capacitance amongst the four amplifiers, but also couples them left to right because the outputs swing simultaneously and this leads to the observed pedestaling of the E_t signal. Fixing this problem, while difficult due to the extreme layout constraints of the FOE chip, seems possible, although not essential since it is small enough effect to not impact the weighting function sufficiently to prevent operation.

On the left of Figure 5-39, we show the image of the gradient magnitude channel, and on the right is the result of the gradient thresholding. The situation is the same as before, with frontal motion. Unfortunately, the gradient channel has the smallest of output signals, and hence uses the largest gain resistor. Compensation requirements resulted in only 7 bits of effective range out of this channel. Resistors larger than this resulted in settling times that were too slow. However, even with only 7 bits of gradient magnitude range the gradient thresholding was quite effective in practice.

When it came time to close the chip in a simple feedback loop, the last and most fundamental error in the chip was discovered. The finite differences used for the image brightness gradients were laid out with x axis direction proceeding conceptually from left to right. In actual fact, the position encoder strobes out voltages which go from low to high. This means that the position encoded results in an x direction which is from *right to left.* Hence, the estimated values for E_x

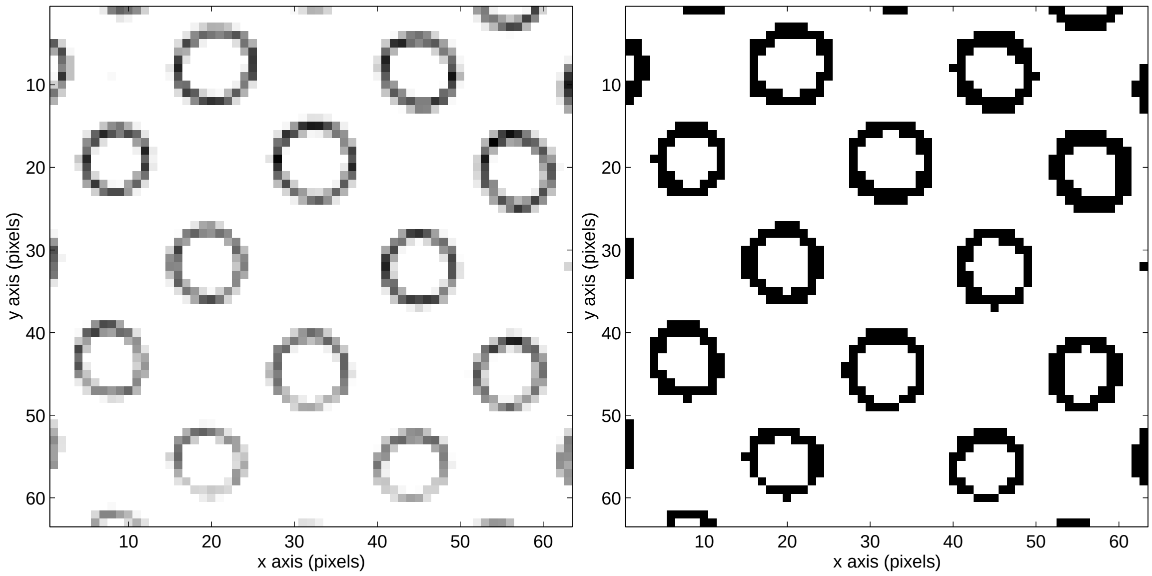


FIGURE 5-39: The image of the gradient magnitude channel during frontal motion is shown on the left, and on the right is the binary image representing the result of applying the gradient thresholding.

have a sign error. The various tests of the all of the processors we performed in such a way as to not make this error obvious. Having a sign error in the E_x channel results in the formation of a dot product of $-E_x(x - x_0) + E_y(y - y_0)$ instead of the desired $E_x(x - x_0) + E_y(y - y_0)$. This kind of error has the effect of mirroring the gradient about the y-axis. The FOE algorithm can be made to converge with this error under certain specific conditions of symmetry about the y-axis in the image, but deviates from the correct output when this symmetry is broken. This was of great puzzlement in the lab until the error was found. The sign problem was discovered by looking at the resulting error directions provided by the error channels and comparing them to what the raw image data predicted. Fixing the problem in a new chip can be easily done by switching the direction of the shift register in the position encoder, or altering the gradient stencil appropriately.

In order to avoid a re-fabrication of the chip, a fix was found to correct the sign error. Altering the signs of the gradients is of course impossible. However, some of the position information is available. We could perhaps change the sign of $x - x_0$. Conveniently enough, the output strobe position x is driven off-chip and is available for us to use. By taking the output for x_0 from the DAC and forming a new x_0' to input to the chip, we can effectively eliminate the sign error. If we perform:

$$x_0' = 2x - x_0 \tag{5.28}$$

and input that to the chip as the current estimate, then the dot product becomes:

$$\begin{aligned} p &= -E_x(x - x_0') + E_y(y - y_0) \\ &= -E_x(x - 2x + x_0) + E_y(y - y_0) \end{aligned}$$

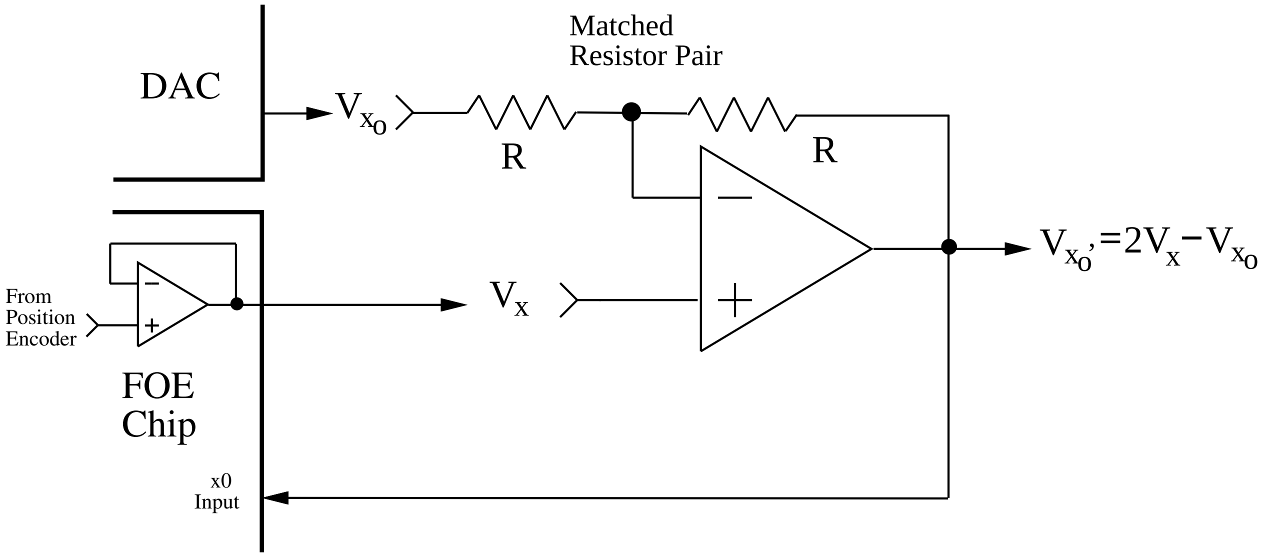


FIGURE 5-40: Simple op-amp circuit for correcting the sign error on the FOE chip.

$$= \quad E_x(x - x_0) + E_y(y - y_0) \tag{5.29}$$

which is the desired result. The simple circuit for implementing this idea is shown in Figure 5-40. The output of the x-channel is still negative, but this sign can be changed quite easily in software.

Once the sign error was corrected, the FOE system began to function properly. Figure 5-41 and 5-42 show mesh plots of the output error channels during a frontal motion transient, where the FOE estimate was held at the image origin. The test board was programmed to perform a simple constant gain feedback loop. An example of the transient response of the loop is shown in Figure 5-43. No optimization of the feedback loop was performed, and the output settling times due to the gains used were typically on the order of 30 iterations, although this is a function of the overall brightness. Proper setting of the gain should be done in the future using the quadratic channel as this has an automatic gain control effect.

Now, with the chip functioning correctly in the feedback loop, we can perform again the same motion experiments that we did when we acquired raw image data. Figure 5-44 shows the transient results for two different orientations overlaid on top of the raw image data that had been acquired previously. The image gradient directions shown are estimated from the raw data, but the selection is based on the processed data from the chip. The first image corresponds to a purely frontal motion, whereas the second image corresponds to a distal positioning of $\theta = -18°$ and $\phi = -18°$.

To test the algorithm performance with the FOE chip doing the processing and compare the results with both the predicted location from the calibration and the location found by

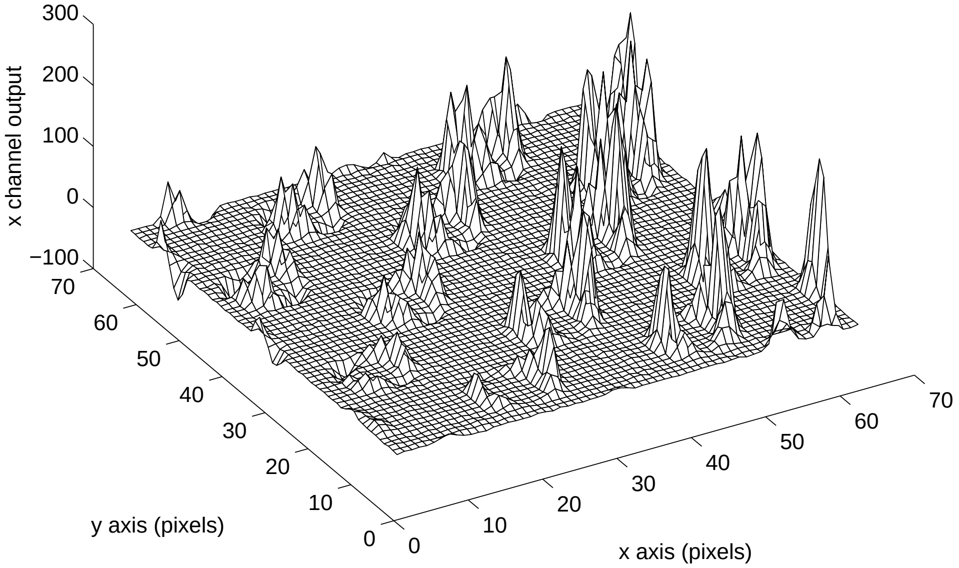


Figure 5-41: Mesh plot of the x error channel output. Motion is frontal, and the FOE estimate is at the image origin.

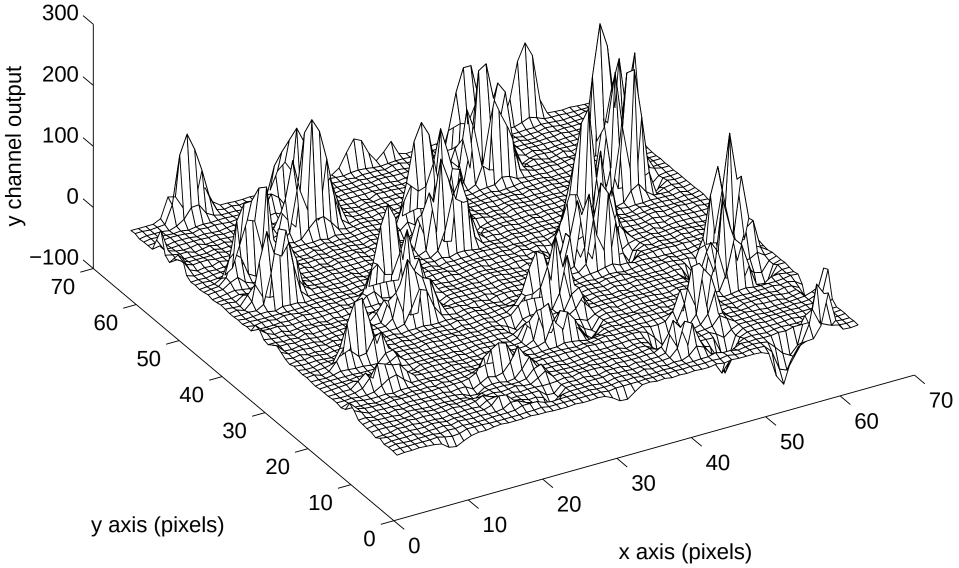


Figure 5-42: Mesh plot of the y error channel output. Motion is frontal, and the FOE estimate is at the image origin.

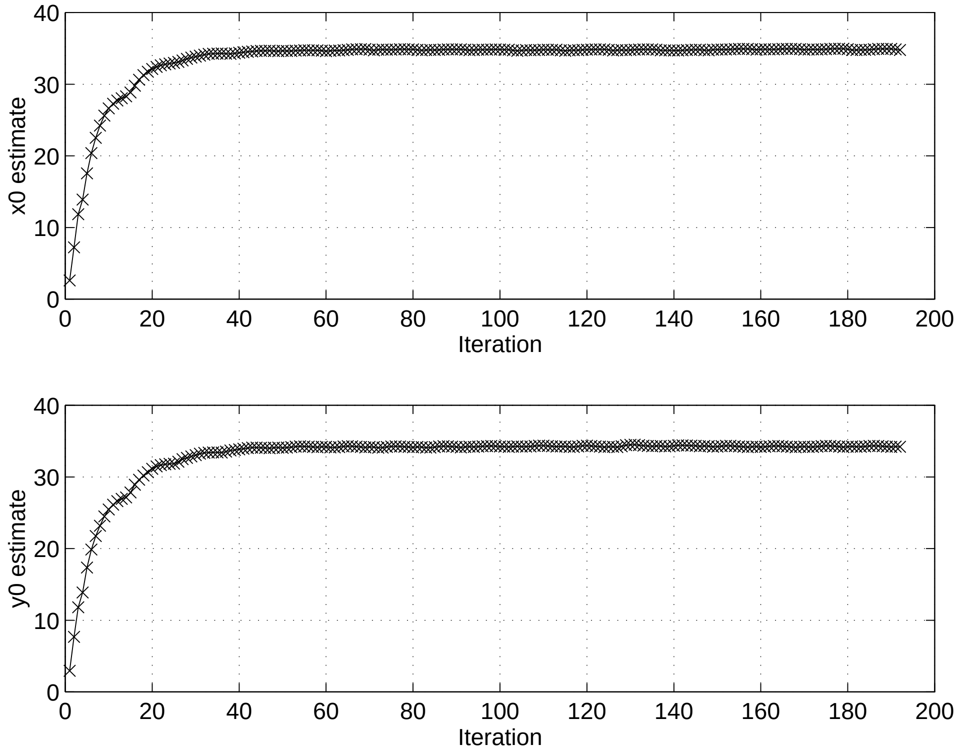


FIGURE 5-43: Transient response of the FOE chip and feedback loop. Motion was frontal, and the estimate started at the image origin.

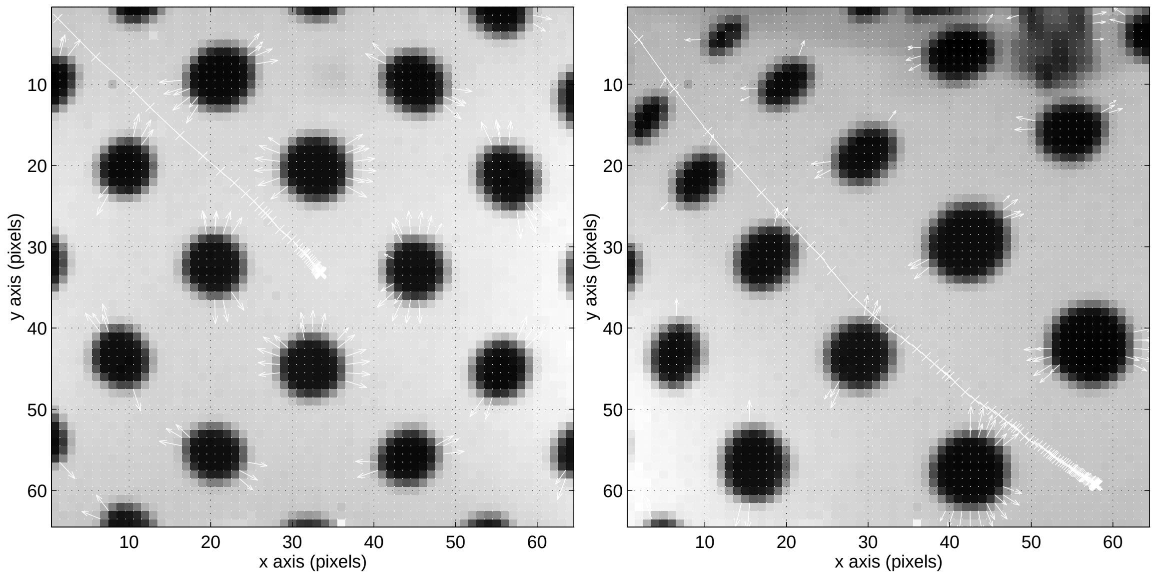


FIGURE 5-44: Example of the algorithm in action using the FOE chip in a simple constant gain feedback loop.

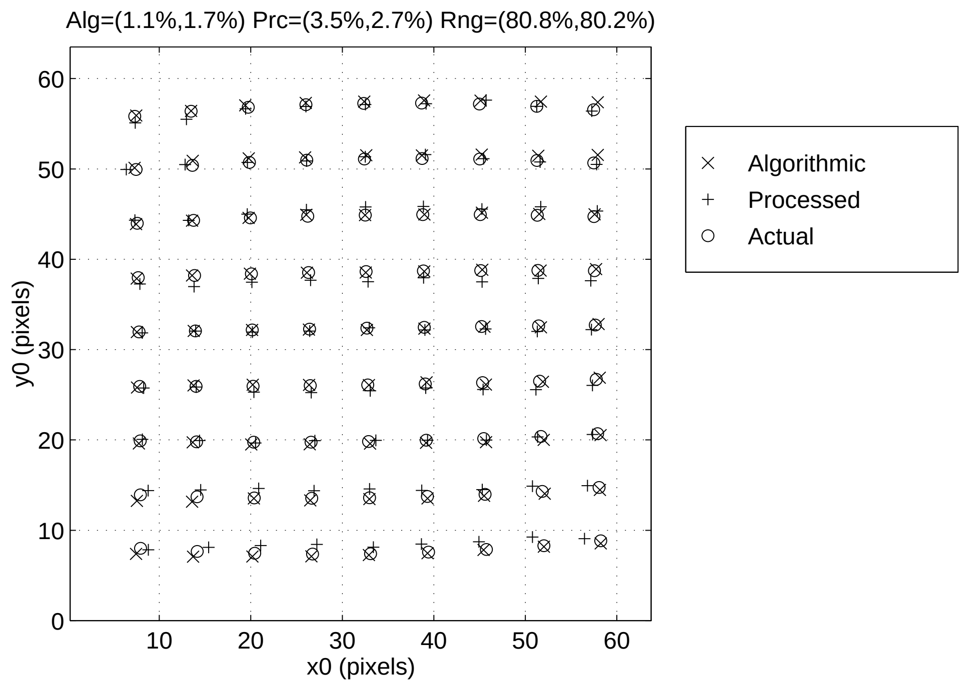


FIGURE 5-45: Comparison of distorted FOE locations (o's) with the results of algorithmic processing of raw image data (x's) and the estimates provide by the FOE chip closed in the feedback loop (+'s).

the algorithm performed on the host computer using the raw image data, the FOE was strobed over the image plane and the resultant processed data from the motion transients was stored on-board. Only 30 frames of data could be stored in such a fashion, but the algorithm was allowed to continue on even after on-board memory was exhausted. Each experiment involved setting the distal mount angles (θ, ϕ) appropriately, and running the stage back and forth for 128 image pairs. Image acquistion was synced with the motion, which was repeated until 128 frame pairs had elapsed, after which all of the information regarding the state of the feedback loop over the motion transient was transmitted to the host computer. Time periods during acceleration and deceleration were, of course, eliminated. For each transient corresponding to a experiment with a particular setting of the angles in the distal mount, a mean location of the FOE was computed. The results of these experiments are shown in Figure 5-45. The distorted locations of the FOE as provided by the camera calibration are shown as o's, while the results of the algorithm in the host computer operating on raw image data are shown as x's. The results of the FOE chip closed in the feedback loop are shown by the +'s. The FOE chip estimated the distorted FOE locations to within 4% over 80% of the field of view. Outside of this range, the situation discussed in Figure 5-33 occurs and the errors increased dramatically.

6

Conclusions

6.1 Summary

This thesis investigated the potential of using 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 embodied in the brightness change constraint equation was developed. Various algorithms for estimating the FOE based on this constraint 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. A comprehensive system for testing and evaluating the chip performance was constructed. This system included all of the various support components needed to operate the FOE chip and to close the required feedback loop around the chip, as well as a calibrated optical/mechanical setup to provide the chip with images corresponding to real motion in real time. The overall chip specifications and test system parameters are shown in Table 6-1. The FOE chip was able to recover the FOE location, and hence the camera motion direction by way of the imaging parameters provided by the camera calibration, to within 4% over 80% of the field of view.

6.2 Improvements and Future Work

During the course of this project, various improvements and potential avenues for future work came to light. They can be broadly categorized in nature as algorithmic-based, chip implementation-based, and testing based.

Algorithmically, the most significant drawback of our framework is the restriction that all

Process	Orbit 10V n-well $2\mu m$ CCD/BiCMOS
Chip Dimensions	9200μm $\times$ 7900μm
Imager Topology	64$\times$64 interline CCD
Technology	double-poly buried channel CCD
Required Number Clocks	17 CCD clocks, 10 CMOS clocks
Illumination	Front-face
Image Sensor	CCD gate
Charge packet size	600,000 e^-
Acquisition time	tested to 1 msec
Quantum efficiency	30% @ 637nm
Dark Current	$\leq$ 10nA/cm^2 @ 30°C
Transfer inefficiency	$\leq 7 \times 10^{-5}$ @ 500 kHz
I/O FGA Output sensitivity	2μV/e^-
I/O Follower Gain	0.97 typical
Sensor Nonlinearity	$<$ 0.5% (7 bits) typical
System Frame rate	30 frames/sec.
Peak On-chip Power Dissipation	170 mW
System Settling time	20–30 iterations typical.
FOE location Inaccuracy	$\leq$ 4% over 80% of the FOV

Table 6-1: Summary of FOE chip and system performance parameters.

variation in brightness over time observed by the system come from the motion of the camera. This rules out situations with several objects moving relative to the camera. Removing this restriction requires the segmentation of the images in the motion sequence with respect to different moving objects. Once this is accomplished, an FOE can be defined and estimated for each object. The motion segmentation problem is a high level type of processing and beyond the framework of the chip. Raw data, however, is available from the chip and could be used by a high-level module to perform the required segmentation. Once achieved, the segmentation could be applied to the data through the masking register, and hence the individual motions computed using the FOE chip. Lastly, future projects might expand our framework to include estimation of rotation simultaneously with translation, as well as a chip for estimating time-to-collision.

In the course of testing the FOE chip itself, the following variety of potential circuit improvements became apparent:

- CCD I/O shift register.
  - Separating the I/O register into 8 channels was unnecessary and led to needless complication. The registers should be made into one.
  - Addition of an output gate in the floating gate amplifier would ease clocking constraints substantially, allowing full four-phase clocking in the I/O register.

  - Larger drive transistors need to be included to reduce the resultant output shift from the cascade of two p-followers.
  - The split-gate input structure should be made of surface channel devices to reduce nonlinearity.

- CCD imager.
  - Imaging was done with CCDs with only a typical quantum efficiency of ≈30%, By using photodiodes, we could perhaps improve this to the 80% range. Backside illumination would also improve the efficiency substantially.
  - In the current implementation, no anti-blooming control is provided, and this was a drawback in practice due to specular reflections in the system. Overflow drains should thus also be provided to keep the excess signal from corrupting adjacent pixel values.

- Floating gate amplifiers.
  - In the array of floating gate amplifiers, a layout error was made. Due to spacing constraints, the outputs of the source followers were routed over the floating gates. This leads to a moderate parasitic coupling between adjacent floating gate outputs. The most noticeable and substantial effect is a feed-through of the image onto the absolute value channel. Correcting this error seems feasible.
  - Addition of an output gate in the floating gate amplifier would ease clocking constraints substantially, allowing full four-phase clocking in the interline registers.
  - Currently, once the charge is through being sensed by the four floating gate amplifiers in a row, it is dumped into a diode connected to V_{dd}. Instead, this diode could be used for charge sensing. The masking register can be used to raster out the image data up the column. This would allow for the output of raw image data at the same time that on-chip processing is occurring.

- Gradient measurement.
  - The chip was discovered to have a sign error in E_x. The stencil used for estimation of the E_x assumed increasing x was from left to right. The output of the position encoder gives increasing x from right to left. To correct this, we can either reverse the direction of the shift register in the position encoder or alter the gradient stencils appropriately.
  - In this implementation of the chip, we are image brightness limited. The error signal magnitudes are proportional to the overall image brightness. An improvement of the system architecture could then be to normalize the output differences to the overall level of illumination as was done in [15].

- Absolute value circuit and cutoff weighting function.
  - Moderate offsets were observed in the absolute value channel, reducing the minimum weighting function width to several percent. This necessitated performing the cutoff weighting function off-chip. The weighting function should be enhanced to do offset cancelation. on-chip.
- Squared Gradient Magnitude.
  - Due to a layout error, only half of the offset from the quadratic circuit was actually cancelled. This led to a large offset current in this channel which was compensated for in the FCP. This is easily fixed through proper device sizing.
  - Algorithmically, adding a gradient magnitude threshold into the weighting function substantially improved performance when the FOE is near the edge of the FOV. Adding this feature to the weighting function, also with offset compensation, should be done.
- The masking register can potentially be modified to allow reading back of the results of the cutoff weighting function.
- In the experimental setup, the feedback loop was done entirely off-chip. The next version of the chip should implement the feedback loop on-chip, as well as do the appropriate gain control using the quadratic channel.

The system for testing the FOE was complex as a result of including a great deal of flexibility, which proved to be invaluable. In a less experimental version, the FOE system is envisioned as needing only three chips, presuming that the feedback loop is implemented on the FOE chip itself. The clock sequences still need to be generated, as it seems unlikely that we would be able to generate them on-chip, and hence an additional clock generation chip would be required. Furthermore, the required clock drivers may necessitate a 3rd separate chip, although this is not as clear. Support for raw image data acquisition still would need to be provided, both for camera calibration and focusing, although an automatic focusing scheme could be implemented.

Lastly, the test system was only able to operate the chip at a frame rate of 30Hz. This was due to the limited light signal provided by the image carrier as a result of the small aperture of its lens system. If we remove the constraints imposed by the image carrier, the FOE chip is expected to be able to operate significantly faster, perhaps by as much as several orders of magnitude, of course depending on the available light. Since observed images scale depth with speed, we can scale the lab tests to the equivalent situation with the faster frame rate. The velocities used in the lab were typically $\approx$ 0.3m/s $\approx$ 1.0ft/s, with a target distance of 0.3m $\approx$ 1ft. Scaling this using a 100 times faster frame rate, the equivalent situation has speeds of 60mph and distances of 100ft, which is obviously commensurate with automotive situations.

Bibliography

[1] J.E. Tanner. *Integrated Optical Motion Detection.* PhD thesis, California Institute of Technology, Pasadena, CA., 1986.

[2] J.E. Tanner and C.A. Mead. An Integrated Optical Motion Sensor. In *VLSI Signal Processing II, (Proceedings of the ASSP Conference on VLSI Signal Processing)*, pages 59–76. University of California, Los Angeles, 1986.

[3] A. Lumsdaine, J. Wyatt, and I. Elfadel. Parallel Distributed Networks for Image Smoothing and Segmentation in Analog VLSI. In *Proceedings of the 28th IEEE Conference on Decision and Control*, Tampa, FL, 1989.

[4] J.H. Harris, C. Koch, J. Luo, and J. Wyatt. Resistive fuses: Analog hardware for detecting discontinuities in early vision. In C. Mead and M. Ismail, editors, *Analog VLSI Implementation of Neural Systems*, pages 27–55. Kluwer Academic Publishers, Boston, MA, 1989.

[5] W. Yang. *The Arcitecture and Design of CCD Processors for Computer Vision.* PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, 1990.

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

[7] C.L. Keast. *An Integrated Image Acquistion, Smoothing and Segmentation Focal Plane Processor.* PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, 1992.

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

[9] D.L. Standley. *Analog VLSI Implementation of Smart Vision Sensors: Stability Theory and an Experimental Design.* PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, 1991.

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

[11] J.M. Hakkarainen. *A Real-Time Stereo Vision System in CCD/CMOS Technology.* PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, 1992.

[12] J.M. Hakkarainen and H.S. Lee. 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, 1993.

[13] L.G. Dron. *Computing 3-D Motion in Custom Analog and Digital VLSI.* PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, 1994.

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

[15] C.B. Umminger. *Integrated Analog Focal Plane Processing for Auotmatic Alignment.* PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, 1995.

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

[17] R. Jain. Direct Computation of the Focus of Expansion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(1), 1983.

[18] B.K.P. Horn and B.G. Schunk. Determining Optical Flow. *Artificial Intelligence*, 16(1–3):185–203, 1981.

[19] A.R. Bruss and B.K.P. Horn. Passive Navigation. *Computer Vision, Graphics, and Image Processing*, 21(1):3–20, 1983.

[20] B.K.P. Horn and E.J. Weldon Jr. Robust Direct Methods for Recovering Motion. In *Proceedings of the First International Conference on Computer Vision*, London, England, 1987.

[21] B.K.P Horn and E.J. Weldon Jr. Direct Methods for Recovering Motion. *International Journal of Computer Vision*, 2(1):51–76, 1988.

[22] K.D. Skifstad. *High-Speed Range Estimation Based on Intensity Gradient Analysis.* Springer Series in Perception Engineering. Springer-Verlag, 1991.

[23] I.S. McQuirk. Direct Methods for Estimating the Focus of Expansion. Master's thesis, Massachusetts Institute of Technology, Cambridge, MA., 1991.

[24] S. Negahdaripour and B.K.P. Horn. Direct Passive Navigation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(1):168–176, 1987.

[25] B.K.P. Horn. *Robot Vision.* MIT Press, Cambridge, MA, 1986.

[26] S. Negahdaripour and B.K.P. Horn. Direct Passive Navigation: Analytical Solution for Planes. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Washington, D.C., 1986. IEEE Computer Society Press.

[27] B.K.P. Horn. Parallel networks for machine vision. In P.H. Winston and S. A. Shellard, editors, *Artificial Intelligence at MIT: Expanding Frontiers*, volume 2, chapter 43, pages 530–573. MIT Press, Cambridge, MA, 1990.

[28] S. Negahdaripour. *Direct Passive Navigation.* PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, 1986.

[29] S. Negahdaripour and B.K.P. Horn. A Direct Method for Locating the Focus of Expansion. AI Memo 939, The MIT Artificial Intelligence Laboratory, Cambridge, MA, 1987.

[30] S. Negahdaripour and B.K.P. Horn. Using Depth-Is-Positive Constraint to Recover Translational Motion. In *Proceedings of the Workshop on Computer Vision*, Miami Beach, FL, 1987.

[31] S.J. Decker. A Resistive Fuse for Image Segmentation and Smoothing. Master's thesis, Massachusetts Institute of Technology, Cambridge, MA, 1991.

[32] P.C. Yu, S.J. Decker, H.S. Lee, C.G. Sodini, and J.L. Wyatt. CMOS resistive fuses for image smoothing and segmentation. *IEEE Journal of Solid-State Circuits*, 27(4):545–553, April 1992.

[33] C.A. Mead. *Analog VLSI and Neural Systems.* Addison Wesley, Reading, MA, 1989.

[34] R.S. Varga. *Matrix Iterative Analysis.* Prentice-Hall, Inc., Englewood Cliffs, NJ, 1962.

[35] C.H. Sequin and M.F. Tompsett. *Charge Transfer Devices: Advances in Electronics and Electron Physics.* Academic Press, 1975.

[36] C. Tomovich. MOSIS — a gateway to silicon. *IEEE Circuits and Devices Magazine*, 4(2):22–23, March 1988.

[37] D.K. Schroder. *Advanced MOS Devices.* Addison Wesley, Reading, MA, 1987.

[38] J.D.E Beynon and D.R. Lamb. *Charge-Coupled Devices and their Applications.* McGraw Hill Book Company (UK) Limited, London, England, 1980.

[39] W.F. Kosonocky and D.J. Sauer. The ABCs of CCDs. *Electron Devices*, 23:58–63, 1975.

[40] A.M. Chiang et al. A CCD Matrix-Matrix Product Parallel Processor. In *Proceedings of the 1984 International Solid State Circuits Conference*, 1984.

[41] E.R. Fossum. Charge-Coupled Computing for Focal Plane Image Preprocessing. *Optical Engineering*, 26(9):916–922, September 1987.

[42] A.M. Chiang et al. A 100ns 16-Point CCD Cosine Transform Processor. In *Proceedings of the 1987 International Solid State Circuits Conference*, 1987.

[43] M.H. White, D.R. Lampe, F.C. Blaha, and I.A. Mack. Characterization of Surface Channel CCD Image Arrays at Low Light Levels. *IEEE Journal of Solid-State Circuits*, SC-9(1):1–13, 1974.

[44] C.L. Keast. A CCD/CMOS Process for Integrated Image Acquisition and Early Vision Signal Processing. Master's thesis, Massachusetts Institute of Technology, Cambridge, MA, 1989.

[45] D.D. Wen. Design and Operation of a Floating Gate Amplifier. *IEEE Journal of Solid-State Circuits*, SC-9(6):410–414, 1974.

[46] P.B. Denyer and J. Mavor. Design of CCD Delay Lines with Floating-gate Taps. *Solid-State and Electron Devices*, 1(4):121–129, 1977.

[47] C. Toumazou, F. J. Lidgey, and D. G. Haigh. *Analogue IC Design: The Current-Mode Approach.* Peter Peregrinus Ltd., Stevenage, United Kingdom, 1990.

[48] Y. Tsvidis, Z. Czarnul, and S.C. Fang. MOS Transconductors and Integrators with High Linearity. *Electonics Letters*, 22:245–246, 1986.

[49] A. Nedungadi and T.R. Viswanathan. Design of Linear Transconductance Elements. *IEEE Transactions on Circuits and Systems*, CAS-31(10):891–894, 1984.

[50] P.R. Gray and R.G. Meyer. *Analysis and Design of Analog Integrated Circuits*. John Wiley & Sons, New York, NY, 1984.

[51] N.H.E. Weste and K. Eshraghian. *Principles of CMOS VLSI Design: A Systems Perspective*. The Addison-Wesley VLSI Systems Series. Addison-Wesley, Reading, MA, 1992.

[52] E. Seevinck and R.F. Wassenaar. A Versatile CMOS Linear Transconductor/Square-Law Function Circuit. *IEEE Journal of Solid-State Circuits*, SC-22(3):366–377, 1987.

[53] D.H. Sheingold, editor. *Nonlinear Circuits Handbook*. Analog Devices, Inc., Norwood, MA, 1976.

[54] K. Bult and H. Wallinga. A Class of Analog CMOS Circuits Based on the Square-Law Characteristic of an MOS Transistor in Saturation. *IEEE Journal of Solid-State Circuits*, SC–22(3):357–365, 1987.

[55] B. Gilbert. A Precise Four-Quadrant Multiplier with Subnanosecond Response. *IEEE Journal of Solid-State Circuits*, SC–3(4):365–373, 1968.

[56] B. Gilbert. A new wide-band amplifier technique. *IEEE Journal of Solid-State Circuits*, SC-3(4):353–365, December 1968.

[57] B. Gilbert. A high-performance monolithic multiplier using active feedback. *IEEE Journal of Solid-State Circuits*, SC-9(6):364–373, December 1974.

[58] B. Gilbert. Translinear Circuits. Technical report, Analog Devices, 1981.

[59] B. Gilbert. A monolithic 16-channel analog array normalizer. *IEEE JSSC*, SC-19(6):956–963, December 1984.

[60] A.B. Grebene. *Bipolar and MOS Analog Integrated Circuit Design*. John Wiley & Sons, New York, NY, 1984.

[61] D. Soo and R. Meyer. A Four-Quadrant NMOS Analog Multiplier. *IEEE Journal of Solid-State Circuits*, SC–17(6):1174–1178, 1982.

[62] J. Babanezhad and G. Temes. A 20-V Four-Quadrant CMOS Analog Multiplier. *IEEE Journal of Solid-State Circuits*, SC–20(6):1158–1168, 1985.

[63] S. Qin and R.L. Geiger. A $\pm$ 5-V CMOS Analog Multiplier. *IEEE Journal of Solid-State Circuits*, SC-22(6):1143–1146, 1987.

[64] K. Bult and H. Wallinga. A CMOS Four-Quadrant Analog Multiplier. *IEEE Journal of Solid-State Circuits*, SC–21(3):430–435, 1986.

[65] J.S. Peña-Finol and J.A. Connelly. A MOS Four-Quadrant Analog Multiplier Using the Quarter-Square Technique. *IEEE Journal of Solid-State Circuits*, SC-22(6):1064–1073, 1987.

[66] H.J. Song and C.K. Kim. An MOS Four-Quadrant Analog Multiplier Using Simple Two-Input Squaring Circuits with Source Followers. *IEEE Journal of Solid-State Circuits*, 25(3):841–848, 1990.

[67] J.S. Arora. *Introduction To Optimum Design*. McGraw-Hill, Inc., 1989.

[68] R.S. Muller and T.I. Kamins. *Device Electronis for Integrated Circuits*. Wiley and Sons, 1986.

[69] R. Gregorian and G.C. Temes. *Analog MOS Integrated Circuits for Signal Processing*. John Wiley & Sons, New York, NY, 1986.

[70] Y. Chigusa, K. Fujiwara, Y. Hattori, and Y. Matsuda. Properties of Silica Glass Image Fiber and Its Application. *Optoelectronics – Devices and Technologies*, 1(2):203–215, December 1986.

[71] Kamyar Eshghi. A System Demonstration of VLSI Early Vision Processors. Master's thesis, Massachusetts Institute of Technology, 1992.

[72] Paul Horowitz and Winfield Hill. *The Art of Electronics*. Cambridge University Press, 1987.

[73] James K. Roberge. *Operational Amplifiers: Theory and Practice*. John Wiley and Sons, 1975.

[74] Daniel H. Sheingold, editor. *Analog-Digital Conversion Handbook*. Prentice-Hall, 1986.

[75] Ralph Morrison. *Grounding and Shielding Techniques in Instrumentation*. John Wiley and Sons, 1986.

[76] Walt Kester, editor. *Mixed-Signal Design Seminar*. Analog Devices, Inc., Norwood, MA, 1991.

[77] Eugene Hecht. *Optics*. Addison-Wesley, 1990.

[78] Gideon P. Stein. Internal camera calibration using rotation and geometric shapes. Master's thesis, Massachusetts Institute of Technology, 1993.

[79] J.J. More et al. User Guide for Minpack-1. Technical report, Argonne National Laboratory, 1980.