AN ANALOG VLSI ARCHITECTURE
FOR
STEREO CORRESPONDENCE

Thesis by
Gamze Erten Salam

In Partial Fulfillment of the Requirements
for the Degree of
Doctor of Philosophy

California Institute of Technology
Pasadena, California

1994

(Defended August 24, 1993)

Acknowledgements

I am grateful to many for making this work possible. I will start by thanking the National Science Foundation Graduate Fellowship Office for their financial support.

I am grateful to my teachers, to my advisor Prof. R. Goodman for his guidance and support, to Prof. C. Mead and Prof. P. Perona for their ideas and motivation, to all members of my candidacy and thesis defense committees for their constructive criticism, to all professors, lecturers and teaching assistants who have devoted their time to the powerful instruction I received at Caltech, as well as to all those whose previous work in this field has illuminated my path.

I am grateful to my friends and colleagues who have given me support and encouragement in many ways. My heartfelt thanks go to my friends, co-workers, and supervisors in San Diego for making my transition from "real-life" to graduate school possible, and to Bingul, Janet, Zehra, Eleni, Bahadir, Petros, Danielle, as well as members of my research group and friends in "Carverland" for making life at Caltech colorful, enjoyable and challenging. Special thanks to Ammar for supporting my chip test at MSU.

I am grateful to my family, my late father for introducing me to the life sciences, to my mother for being a super role model with her own career success, to my sister for providing just the right amount of competition, and to my husband Fathi for not only bearing with me through the time that my work was consuming all my energy, but encouraging and helping me do my best.

I thank them all and thank God for having sent them my way.

Abstract

My goal in engaging in this project was to design a hardware system to solve the stereo correspondence problem in real-time.

Consequently, this work describes and analyzes an algorithm for stereo correspondence, its extension to an analog VLSI architecture, and the results obtained from its hardware implementation as a chip.

The first chapter, titled Introduction, describes the stereo correspondence problem. Therein, I discuss biological and psychophysical mechanisms of stereo vision, and include a brief history of ideas to date on the subject. I wrote this chapter to introduce the problem to the reader without assuming any previous knowledge about vision. I believe that reading it with the aid of definitions in the glossary can equip most any reader with information regarding the basics of stereo vision.

The second chapter, titled In Search of the Correct Similarity Measure, expands, first by a simple example, later in mathematical terms, the issues involved in the selection of a similarity measure. The similarity measure is a key component in the solution of the stereo correspondence problem. My main approach is a statistical one, using probability distributions and Bayesian analysis. The chapter motivates the two-sided approach of the algorithm, by using a disparity and a confidence metric for each image region.

The third chapter, titled Simulating the Hardware Algorithm, describes my stereo correspondance algorithm in detail. Simulation results that include both disparity and confidence values obtained with a variety of images are presented. Experiments are conducted

to demonstrate the effect of parameter adjustments. In addition, the algorithm is compared with other correspondence schemes which use various different similarity measures.

The fourth chapter, titled Analog VLSI Implementation, is devoted fully to the hardware implementation. First, the details of the hardware architecture are described. Then, results are presented with two unique implementations. As in the previous chapter, experiments are conducted, this time using the chips themselves. Their results are compared with simulation. Again a variety of images are used.

The fifth chapter, titled Conclusions and Future Work, summarizes the work and explores future expansions.

Table of Contents

List of Figures

List of Tables

Chapter 1 Introduction

1.1 Early vision problems

Confusion arises in discussing early visual processing, particularly if one tries to refer to biological and computational visual systems simultaneously. The biological visual systems are not always organized to process as computational systems do: They often perform computations in parallel or seemingly "out of order." In my description of early visual processing, I will refer to function and purpose only, and not to the organization of biological or computational process.

Early visual processing could be divided into four categories as listed below:

1. Filtering and edge location are processes which transform input image values by a local and/or global computation. This has a three-fold function:

   (a) reduce the amount of information to later stages of processing;

   (b) enhance features that are relevant to later stages of processing;

   (c) reduce the undesirable effects of noise.

   Biological systems start this process at the image formation and sampling stage. The retinal receptors are coupled together by underlying (horizontal) cells to filter the image as it is formed. Other systems that make use of light receptors to capture light have filtering effects of their own due to a variety of physical phenomena. More often, though, these effects are undesirable and need to be minimized.

2. Range transforms infer three dimensional scene geometry from a pair of two dimensional images captured on projection planes of known relative position. A classical biological example of range transforms is stereopsis, the process of determining relative distance of objects in the scene by relating their corresponding positions in the two image planes, or retinae.

3. Surface orientation can be inferred about a surface if its reflectance properties and the light source illuminating the scene are known.

4. Relative motion of the objects in a scene can be inferred from temporal and spatial derivatives between image sequences taken at known intervals.

Early visual processing often utilizes a pyramid for representing copies of visual information at multiple resolutions. A pyramid is a utility structure which can dramatically improve the performance, speed, and robustness of visual processing [BB82]. In biological systems, the functional equivalent is the channels of neural information that communicate and process visual data at different resolutions.

I will explore the second category above at length. I will particularly be concerned with the solutions to **stereo correspondence** which is the process of identifying corresponding image points.

1.2 Stereopsis and stereo correspondence

1.2.1 Problem description

The classical discussion of **stereopsis** begins with the description of two cameras (or eyes) separated by a baseline obtaining slightly different views of the scene. Below are the sequence of steps that constitute the solution to the stereopsis problem:

1. Obtain two images of the scene separated by an appropriate baseline.

2. Select areas containing appropriate image features (targets) to be matched between the two images.

3. Find corresponding target pairs between the images, i.e., match each one in one image with its representation in the other image.

4. Using the appropriate constraints, interpolate between the sparse values of spatial relationships between corresponding target pairs to obtain a dense **disparity** field.

5. Using the position of the two cameras and the disparity field, determine the relative three dimensional coordinates of each point **(depth)**.

These concepts are illustrated pictorially in Figure 1.1. The top plane with horizontal stripes constitutes the scene, which is captured on the bottom two pairs of image planes. The planes on the left illustrate convergent geometry; whereas, those on the right illustrate non-convergent geometry. The stripes themselves are the epipolar lines. Geometrical complexity is significantly reduced by viewing a surface parallel to the baseline (fronto-parallel plane), while the projections are co-planar (non-convergent camera geometry). The biological analogy using two eyes illustrated in Figure 1.2 shows disparity as an angle, providing a more accurate method of definition and measurement. Disparity is equal to the difference between the two angles, or $(\beta - \alpha)$. Although the image matching search space is two dimensional in the general case, the search itself is one dimensional along **epipolar lines.**

The main focus of this study is the third step above, termed the **stereo correspondence** problem.

For more information on the stereopsis problem, the reader is advised to consult Horn [Horn86] and/or Grimson [Grimson81].

1.2.2 Constraints of image matching

Properties of physical surfaces constrain the behavior of surface position, and consequently define the properties that a correct match must possess. These can be written as the constraints of the image matching problem:

1. Compatibility: Image intensities or targets must be compatible in order to match. For binary targets, this constraint implies that a black pixel can only match another black pixel and a white pixel can only match another white pixel. For images with a multitude of intensities and added noise, compatibility is much harder to define or establish. The next chapter will explore various methods of assessing compatibility between two image regions by using similarity functions.

2. Uniqueness: Almost always, one pixel from one image can match no more than one pixel from the other image. Monocular and occlusion regions are the exceptions (Section 1.2.3).

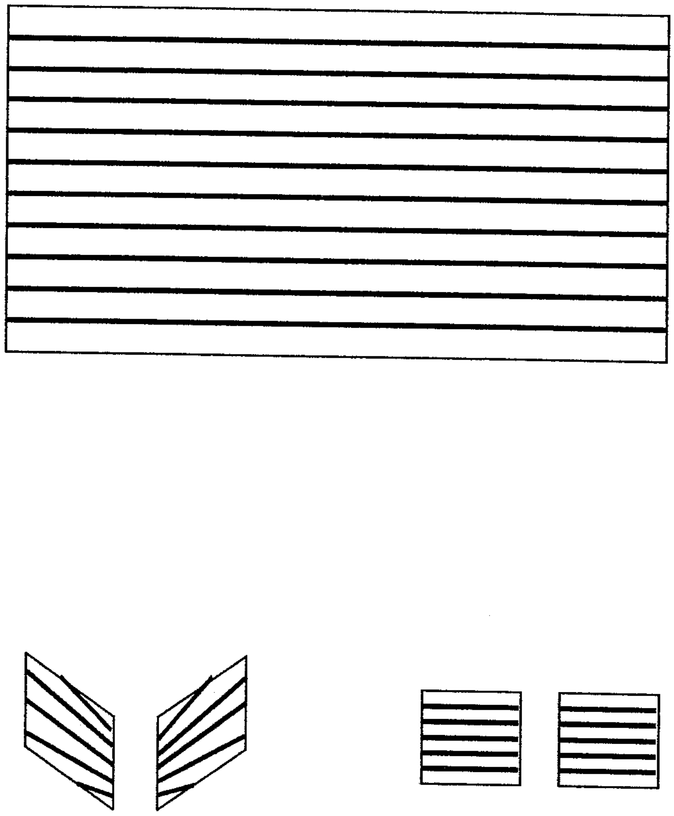


Figure 1.1: Epipolar lines and convergence

The scene is composed of a fronto-parallel surface with horizontal stripes, which coincide with the epipolar lines. Their projections are shown on the two pairs of image planes at the bottom of the figure. Both convergent (left image planes) and non-convergent (right image planes) geometries are illustrated.

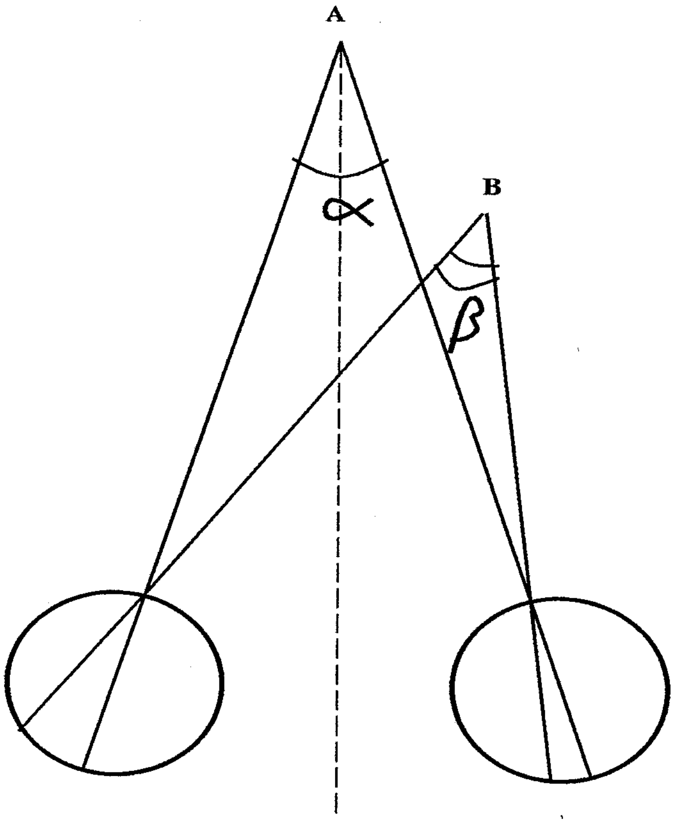


Figure 1.2: Demonstration with spherical image planes, or eyes

3. Continuity: The disparity of the matches varies smoothly almost everywhere over the image. Depth discontinuities, for which occlusion can be an important cue, are the exceptions (Section 1.2.3).

1.2.3 Psychophysics

Classical theories of stereoscopic vision: a brief review

During the Renaissance period painters were aware of the difficulty of articulating a three dimensional world on the two dimensional medium of a canvas. To succeed they employed **monocular** cues to create a sense of distance. They used light and shadow as well as perspective to give a sensation of depth in their works.

Most cues for monocular and **binocular** distance perception were identified by the beginning of the eighteenth century. What remained most puzzling, however, was the singleness of vision: How images from two eyes resulted in one image. The investigation of this issue ultimately led to the discovery of the cue of retinal disparity [GL76].

Any history of stereoscopic vision must mention the **stereoscope** and its inventor C. Wheatstone. The significance of this mid-nineteenth century development is that it represents the first clear demonstration of the fact that one can perceive depth as a result of dissimilar retinal images.

Later contributions by P.L. Panum elaborated on the notion of **stereo fusion**. He determined that fusion occurred when images fell within a certain distance on the two retinae. He experimentally found the limit to be 0.052mm, or the width of 15 to 20 **cones**. He called these areas "corresponding circles of sensation" [GL76]. This was in close agreement with the earlier postulate of J. Müller which geometrically illustrated the loci of points stimulating identical retinal points between the two eyes (**Müller's horopter**). Before the end of the nineteenth century two other German psychologists, Hering and Helmholtz added to the existing theories by noting the role of experience, as well as that of attention.

With the twentieth century came the predominance of **Gestalt psychology**. There was a more global approach to stereo vision, which in effect claimed that monocular form of

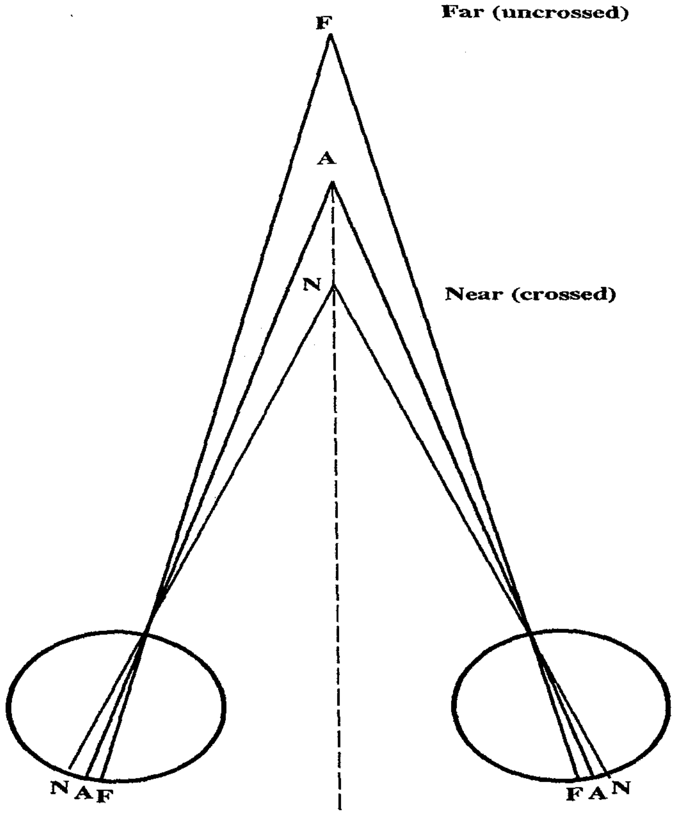


Figure 1.3: Crossed and uncrossed disparities

recognition was essential for depth perception. This idea was later shown to be incorrect by Julesz whose **random dot stereograms (RDS)** demonstrated that depth perception can occur in the absence of monocular cues or **contours** [Julesz71].

Two main cues for stereopsis: disparity and occlusion

The significance of disparity was realized quite early in the history of ideas related to stereopsis. One may notice even as a child that blinking one's eyes alternatively makes an object held close to the face "jump" from right to left and vice versa. This is a simple illustration of the disparity cue. Points closer than the horopter have crossed (negative) disparity; whereas, points further than the horopter have uncrossed (positive) disparity (Figure 1.3).

When viewing non-fronto-parallel planes, disparity takes on a more interesting form. A one dimensional picture shows a "frequency modulation" type effect. This leads to the concept of **disparity gradient**, the rate of change in disparity as one moves across the scene.

The second major cue for stereo vision is **occlusion**, first noted, though rather indirectly, by Leonardo da Vinci in his description of "Leonardo's parallax." Later Helmholtz elaborated on the role of occlusion in defining **stereoscopic contours** by integrating it to his concept of "unconscious inference." Although this concept offers little by way of explanation today, it was his method for describing the role of occlusion in unconsciously grouping stereoscopic contours. Figure 1.4 shows an experiment which illustrates that occlusion is essential for such a grouping [GL76]. The pair of rectangles on the top of the figure lead to the sensation of two surfaces, a white closer surface occluding a farther surface of black dots. The bottom pair, on the other hand, creates the sensation of two frames - not surfaces - one closer and the other farther from the viewer.

Further experiments show that stereoscopic contour surface edges are perceived not only at occlusion points but at edges of sets of points suggested by occlusion to belong to the same stereoscopic contour (Figure 1.5)[GL76].

As one could guess, the notion of occlusion remains more elusive and difficult to understand and incorporate to a stereopsis algorithm than that of disparity.

1.2.4 Neurophysiology

It is the plight of neural science that its quest for understanding neural function has traditionally begun at the cellular level. This is similar to trying to determine the operating principles of a supercomputer by examining individual transistors! Stereo correspondence which was an early subject of research has not escaped this fate. Initial probing to search for "stereopsis cells" were carried out in the **primary visual cortex**, which is the first point along the visual pathways where the nerve signals from the two eyes come together. These studies hinted the presence of cells narrowly tuned to specific disparities in the cat

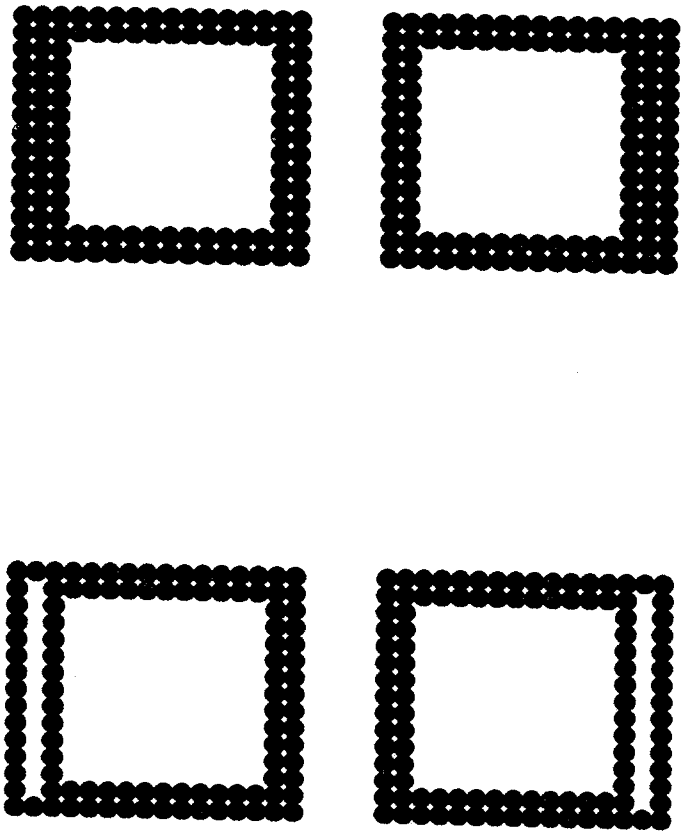


Figure 1.4: Importance of the occlusion cue for separating surface contours

The pair of rectangles on top lead to the sensation of two surfaces, a closer white surface occluding a farther surface of black dots. The bottom pair; on the other hand, creates the sensation of two frames - not surfaces - one closer and the other farther from the viewer.

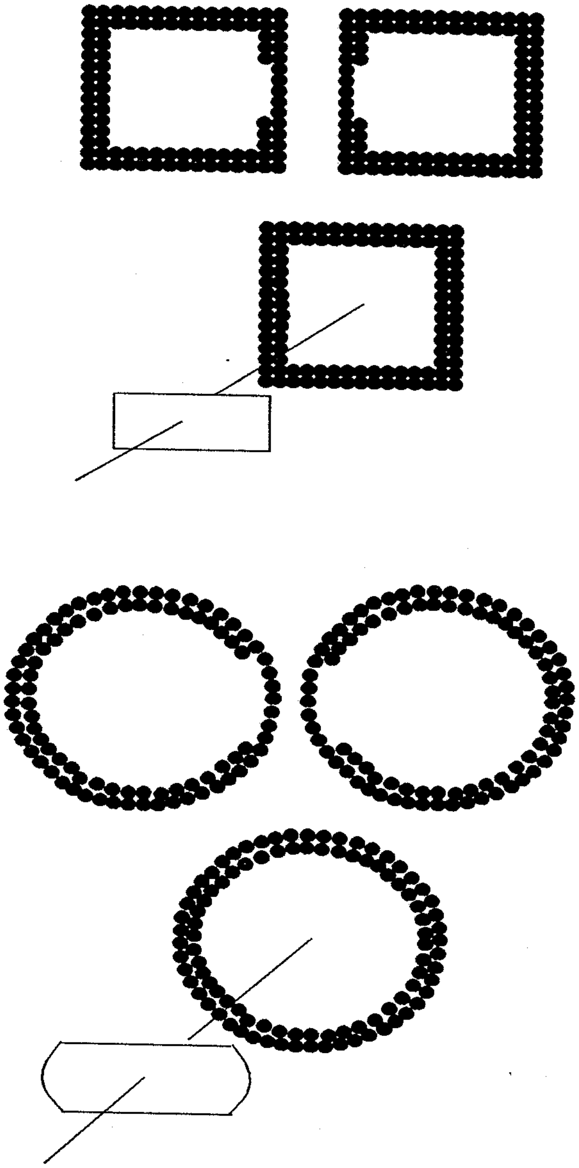


Figure 1.5: Occlusion as a cue for creating surface contours

The top pair of images leads to the perception of a rectangular white surface closer to the viewer. The bottom pair creates another rectangular surface with arcs forming the width boundaries.

[BBP67]. So scientists were led to believe they were "disparity detectors." Subsequently, repeated experiments determined that the visual cortex does contain disparity tuned cells and most of them prefer disparities of half a degree or less [PF77] [CW77]. Today the term "detector" is considered rather misleading because it creates the false presumption of a binary system that defies continuity and robustness, the two main attributes of neural computation. Besides, depth perception relies on cues in addition disparity (Section 1.2.3). Perhaps it is preferable to call these cells "stereo analyzers." Their two main categories are listed below:

1. Near-zero disparity tuned cells respond to stimulus in the immediate region of the fixation plane. Most of these are "tuned excitatory," meaning that they respond to near-zero disparities by increasing their firing rate. The rest are "tuned inhibitory," i.e., they are silent in response to the same.

2. Near and far cells respond in an excitatory manner over a wide range of disparities of one sign and are inhibited by disparities of the opposite sign.

Figure 1.6 illustrates the response of these cells pictorially. It is believed that depth perception arises from the relative responses of these pools of cells [RFPST90].

The mechanisms and the selection of targets for the stereo matching process are most likely embedded in the inputs to these cells, as well as in the architecture of their connectivity. Anatomical and psychophysical evidence suggests that at least several channels participate in the matching process [JM75]. Also, we know that both of the two main visual pathways (magnocellular and parvocellular) participate in stereo vision [KSJ92]. As for the mechanism, many believe that given the nature and constraints of the computation (Section 1.2.2), a locally connected, globally interacting architecture, utilizing local correspondence primitives is implicated. Local correspondence primitives are the similarity measures between neighborhood regions of the two retinae. The search inside these neighborhoods determines the corresponding points. The size of the search regions has been determined to vary with the frequency of the signals that make up the images. Maximum fusible disparity scales with the spatial frequency of the stimulus [Marr82].

The image matching system is robust: It is immune to head and eye movements. Its

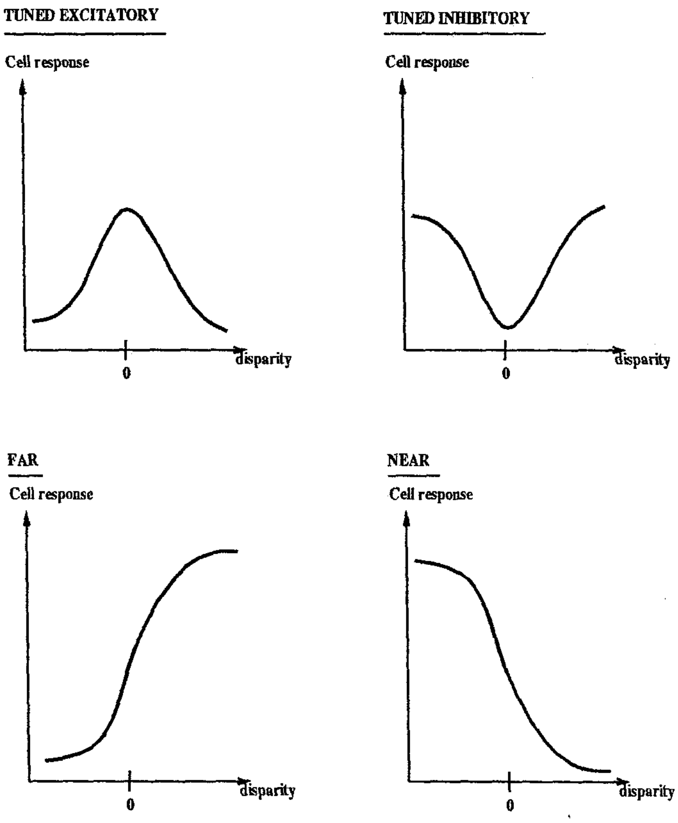


Figure 1.6: Responses of disparity tuned cells in the visual cortex

ability for matching extremely fine features is impressive: **Hyperacuity** is reported to be 2" of an arc ($\frac{1}{1800}$th of a **degree of visual field**), corresponding to about $\frac{1}{12}$th the diameter of a **foveal** cone [Berry48]. The ideas about the mechanisms for accomplishing such formidable tasks remain speculative.

1.2.5 Computational approaches

I already mentioned the steps that constitute the stereopsis problem in Section 1.2.1. If one wanted to design an algorithm to accomplish each step on that list, how would one go about it? One approach is to study neurophysiological and psychophysical data from biological visual systems and model their approach [Marr82]. Another is to analyze the properties of image signals and devise a scheme to maximize the probability of a successful match [Weng90] [OK89] [JM92]. A third more implementation-minded approach is to tailor a method specifically for a physical, technologically achievable medium of real-time computation and integrate the above two approaches as much as possible [MD89] [Mahowald92]. My objective is similar: Solve the stereo correspondence problem in the medium of integrated circuits, or **VLSI**.

Whatever their approach, most stereopsis algorithms have certain defining attributes. Since the focus of this study is stereo correspondence, I will outline the major categories in which image matching algorithms differ:

1. Target selection: Identifying the correct points while minimizing the number of false matches is the objective of this selection process. The targets can be sparse, such as edges or the zero crossings of the Laplacian of a Gaussian [MP79] or areas identified by an interest operator. Alternately, they can be dense, such as the intensity values themselves [Barnard86], windowed Fourier phase [Weng90] or an appropriate filtered version of the two images.

2. Similarity measure: The objective is to select the best criterion that signals correspondence between two targets. Similarity measure must also accommodate the constraints of stereo matching. For binary edge targets, orientation or sign of the edge can be used to enhance the probability of a correct match [Grimson81]. Correlation [Hannah74] [WTK87] and intensity difference [OK85] are two popular similarity

measures with dense targets.

3. Single versus multiple resolution: Image matching can be more likely to produce the correct solution if the search is conducted at more than one scale of resolution. To clarify, consider two versions of the same image, one blurred and the other sharp. The ambiguities in matching the targets in the sharper image can be resolved by matching the blurred image. This is roughly what the coarse-to-fine matching algorithm is about [Grimson81]. Some algorithms use multiple resolution in tandem with multiple target selection schemes by employing multiple filters for each resolution [JM92].

4. Local versus global: This refers to the nature of the computation. Global computation evaluates the validity of all target matches in the image, at least those lying along one epipolar line, simultaneously. Local computation, on the other hand, matches one area of the image at a time, paying little if any attention to other areas. Serial algorithms which utilize windows [Hannah74] are local; whereas, parallel algorithms which allow for cooperative schemes between regions of the entire image [MP79] are global. One might conclude, considering the constraints (Section 1.2.2), that necessity of global computation is implicated. However, multiresolution algorithms such as coarse-to-fine matching strategy, locally acting constraints such as disparity gradient limit [PMF85] and iterative dynamic methods such as Kalman filtering [MKS89] greatly enhance the ability of local computation to produce the correct match.

1.2.6 Solving the stereo correspondence problem in hardware

Many robotics and navigation tasks using stereopsis require real-time computation, necessitating direct physical implementation of the image matching algorithm. In addition for autonomous behavior they require small-size, low-power computing systems to handle the enormous information associated with image processing. The challenges lie in the following areas:

1. Managing communication between image sensor and image processor arrays: One way to bypass this bottleneck is to integrate the two. But to find a physical medium with the right photoelectric properties, capable of fast, reliable computation and producing realistic image size and resolution is no simple task. Biological vision

systems, which separate the image formation task from most visual processing tasks, face the same data-reduction challenge.

2. Parallelizing the computation: In real-time processing there is little chance to refine raw output. The matching algorithm must produce the best result at first pass. This is especially difficult if hardware does not provide enough area to handle the whole image pair concurrently.

3. Simplifying the computation: We know that the computation of most functions in hardware are implemented by iterative methods. Real-time processing restricts this capability so a hardware matching algorithm must utilize simple arithmetic or arithmetic that is easily computable within the physical medium.

Conventional approach to image processing in VLSI has been micro-programmable systolic array implementations [Parker85] [WK86]. There have also been attempts towards parallelizing the computation by pipelining [ITMMSH86]. Yet, these still remain inadequate for real-time computation of most early vision problems. Predictably, stereo correspondence is among them with its enormous demand for data space and computational density.

A newer hardware approach to visual processing has been to use analog VLSI processing arrays [SMM87]. This approach has matured over time to tackle early vision problems such as retinal adaptation [Mead89a], motion [HKLM88], color constancy [MAG91] and most recently stereo correspondence problems [MD89] [Mahowald92].

My study expands on the same tradition by implementing a hardware stereo correspondence algorithm to handle two dimensional images serially.

Chapter 2
In Search of the Correct Similarity Measure

2.1 Motivation for a statistical analysis

Images that surround us (fortunately) do not look like an untuned television screen, but contain distinguishable regions of smoothly varying intensity values [Horn86] [VK92].

One major current trend in the analysis of the regularities in images has been to note their *nondeterministic* nature by modeling correlations and likelihoods. Lattice-based random field models and spatial statistics have been promising tools in capturing and quantifying the regularities that enhance the robustness and success of many image processing algorithms [GG91] [Chen88].

This chapter contains analysis of the image matching problem. I think it will be interesting to start with a simple example to illuminate the concepts encapsulated in the equations in the following sections. To this end, I designed the following matching example.

2.2 A simple matching example

Consider two images of a scene made up of square areas or **pixels**. These pixels can take on only three values, white, gray and black. Black and white pixels are equally probable, but gray ones are twice as likely as black or white pixels. Probability of having a particular pixel is equal for both images. Image matching is carried out between the two images by the following procedure, also illustrated in Figure 2.1:

1. Pick one pixel from the first image.
2. Pick the pixel in the same location as that in step 1 and four neighboring pixels from the second image, thus forming five candidate pixel pairs.
3. Select the pair that contains the correct match.

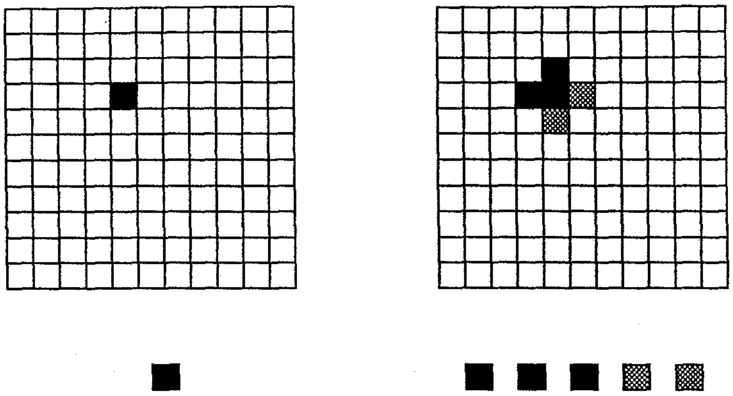


Figure 2.1: Example matching procedure

There are five candidate pairs of pixels for each match, and exactly one of these is the correct match. A complete list of all possible pixel pair candidates are shown in Figure 2.2. These choices are consistent with the assumptions made to this point. The choices are assumed to be limited to this set for simplicity.

One immediate observation is that elements of the pixel pairs are not independent:

$$P(XY) \neq P(X)\,P(Y) \tag{2.1}$$

The exact values of $P(XY)$ are listed in Table 2.1.

Now examine the probability of a match given a particular pixel pair (XY), i.e., $P(M|XY)$. The exact values are shown in Table 2.1.

Although without prior knowledge of image statistics we have no means of refining our decision given identical pairs such as (GG) and (BB), the example shows that

$$P(M|GG) \neq P(M|BB) \tag{2.2}$$

Intuitively, one would expect identical pixels to match with equal probability. The distinction is embedded in the concept of a **match**:

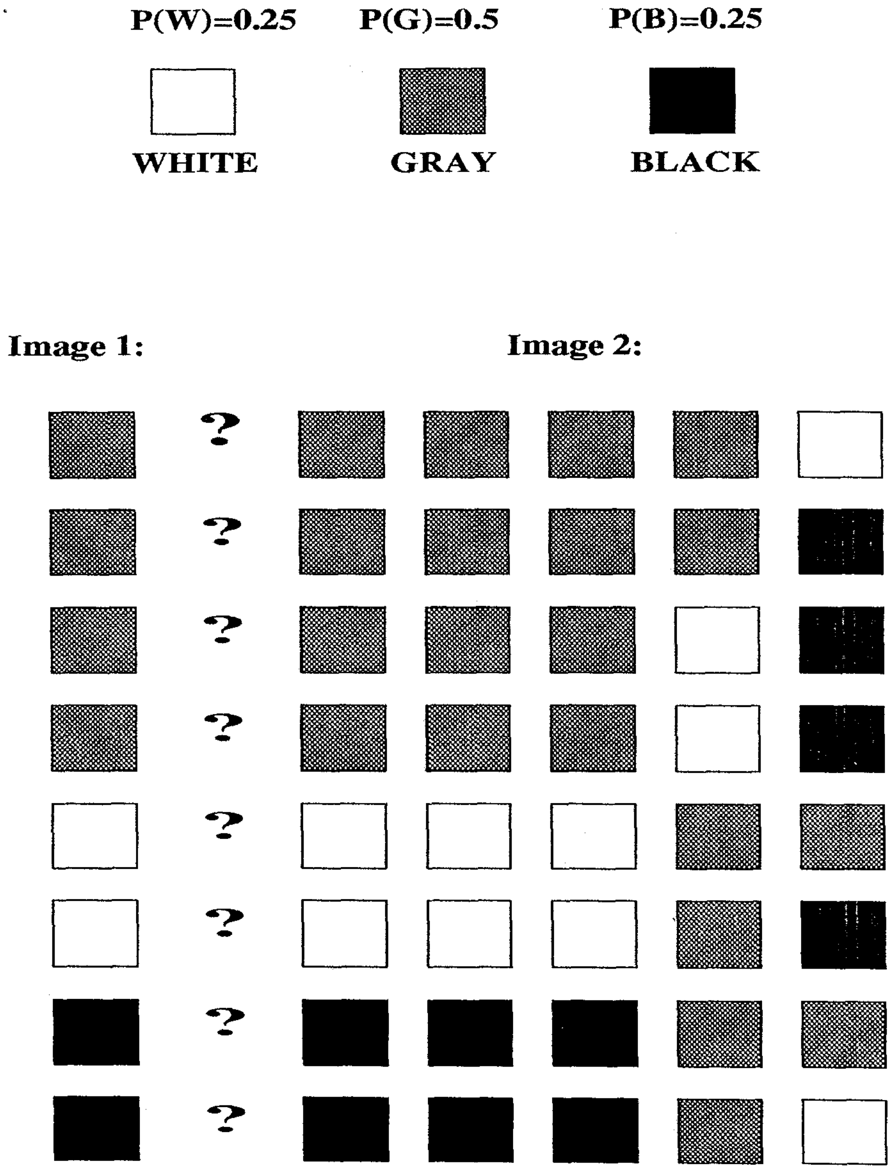


Figure 2.2: Example matching statistics

Pair	Probability
GG	$\frac{7}{20}$
WW	$\frac{3}{20}$
BB	$\frac{3}{20}$
GW,WG	$\frac{3}{40}$
GB,BG	$\frac{3}{40}$
BW,WB	$\frac{1}{40}$

Event	Conditional probability
$P(M_{actual}\|GG)$	$\frac{2}{7}$
$P(M_{actual}\|WW)$	$\frac{1}{3}$
$P(M_{actual}\|BB)$	$\frac{1}{3}$

Event	Conditional probability
$P(M_{possible}\|GG)$	1
$P(M_{possible}\|WW)$	1
$P(M_{possible}\|BB)$	1

Event	Probability
$P(M_{actual})$	0.2
$P(M_{possible})$	$\frac{13}{20}$

Table 2.1: Table of example matching statistics

To this point, we only considered the set of actual matches or $\{M_{actual}\}$. It is helpful to define a second set of $\{M_{possible}\}$ where $\{M_{actual}\} \subseteq \{M_{possible}\}$ and

$$P(M_{actual}) \leq P(M_{possible}) \tag{2.3}$$

Such a relationship arises from the ambiguous nature of the image matching problem. In regions where the $\{M_{actual}\} \equiv \emptyset$ (i.e., monocular or occluded regions)

$$P(M_{actual}) = 0 \leq P(M_{possible}) \tag{2.4}$$

Most decision rules that assign pixels (or a region of pixels) in one image to corresponding pixels (or a region of pixels) in the other image utilize metrics that do not consider image statistics. Such metrics usually base $P(M|XY)$ solely on the output of a **similarity function** $f_m(X,Y)$ yielding the same value for identical matching pairs:

$$f_m(X,X) = f_m(Y,Y) \tag{2.5}$$

The above assumption implies that

$$P(M|XX) = P(M|YY) \tag{2.6}$$

It is clear that the event M in Equation 2.6 could replace $M_{possible}$, but can not replace M_{actual}. Consequently, a simple similarity function, such as the one in Equation 2.5, can not contain a complete statistical solution. Three of the similarity functions that will be considered in Section 2.3.2, namely, the difference metric, the difference squared metric and the hardware metric share the property shown in Equation 2.5. The other two, the correlation and the normalized correlation metrics do not. However, the dependence of these two on the exact values of X and Y does not reflect a compensation for image statistics. Therefore, assessment 4 below holds for all five metrics I will discuss.

This simple example illustrates some of the important aspects inherent to image matching:

1. Intensity values in images are not uniformly distributed: The trend in the literature is to use normal distributions, if not for the whole image, for regions in the image [Wesseley76]. To exemplify the choice of $f_x(x)$ in this analysis, image intensity value distributions from a natural image (photograph of an outdoor scene, see Figure 3.16) are shown in Figure 2.3. The top plot was obtained without any processing. The bottom plot was obtained after the image was filtered using a Gaussian kernel. Note that x is a single pixel intensity value (i.e., a scalar).

2. Local spatial correlation leads to significant cross-correlation between image pairs: The trend in the literature is to use joint normal distribution with correlation coefficient ρ_i (image correlation coefficient) [Wesseley76].

3. A mechanism for resolving ambiguities is needed: The *perfect* match need not be the *correct* match. One needs to utilize the constraints of the matching problem to reduce this inherent ambiguity.

4. A similarity function alone can not provide a complete statistical solution to the image matching problem.

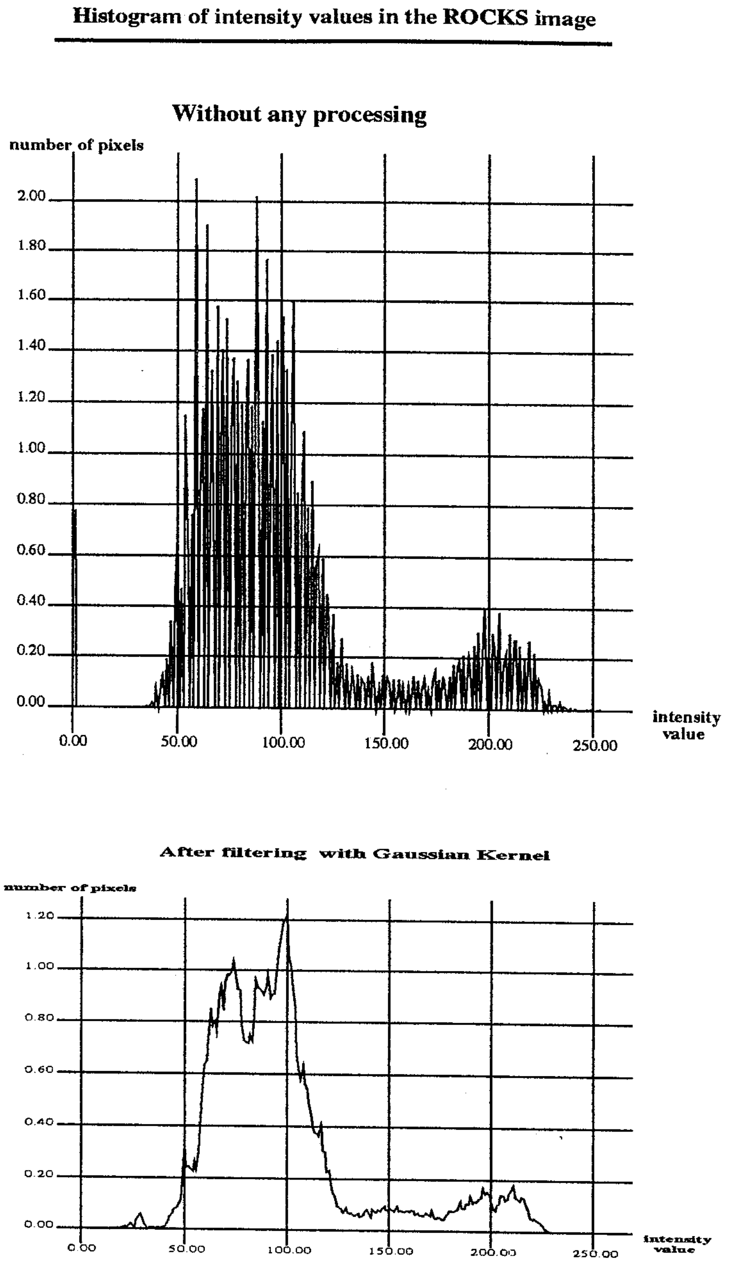


Figure 2.3: Histograms of gray level distributions in a natural image

The initial image distribution contains many singularities. There are many gray levels that are not present in the image (i.e., number of pixels equals zero) even though immediately adjacent intensity values are highly likely. The singularities can mostly be attributed to the digitization process. After processing with a Gaussian kernel of high σ, the intensity value distribution contains far less singularities and many more piecewise continuous regions. This process also increases the coefficient of correlation in the processed image. Note that the vertical axes is scaled by 1000.

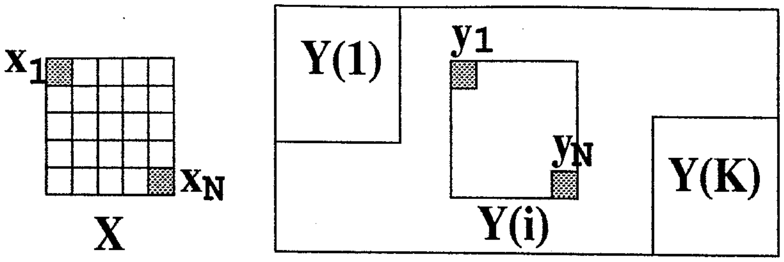


Figure 2.4: Image matching procedure for vectors

2.3 Statistical approach to image matching

2.3.1 Probability distributions and similarity measures

Assume that correspondence between two images is going to be determined by taking a region in one image and searching an appropriate neighborhood in the other image for the corresponding region (Figure 2.4). Disregard for the moment the issues involved in the choice of the size of the region and the size of the search space. (An excellent discussion on these is given in [OK89].) Define a region X in the first image, a vector of length N, composed of the sampled pixel values x_i, and search region $\{Y_i\}$ in the second image, which is a set of vectors exactly one of which is a true match for X. (We are ignoring occlusion and monocular regions by this assumption.) Which vector in the set $\{Y_i\}$ maximizes the probability of a match $P(M|X)$? This is a problem to which one can apply Bayes Rule that relates *a priori* probabilities $P(X)$ and $P(M)$ with *a posteriori* probability $P(M|X)$ and conditional probability $P(X|M)$:

$$P(M|X) = \frac{P(X|M)P(M)}{P(X)} \tag{2.7}$$

We can replace M above with $\overline{M}$ and obtain the analogy for a no-match. Since

$$P(X) = P(X|M)P(M) + P(X|\overline{M})P(\overline{M}) \tag{2.8}$$

one can obtain the following interesting ratio for the *a posteriori* probability:

$$P(M|X) = \frac{1}{1 + \frac{P(X,\overline{M})}{P(X,M)}} \tag{2.9}$$

For simplicity the above equations were derived assuming discrete values for the elements of region X, so that $P(X)$ represents a finite probability. Also, the event match (M) is binary: It either occurs or not. If image values are continuous, $P(X)$ will be replaced by the respective probability density $f_X(X)$. In that case, $P(M|X)$ would no longer be a discrete quantity but a probability distribution itself, $f_M(M|X)$. Consequently, we face a much more complicated situation where we must rank not discrete numbers but probability distributions.

Although in the discrete simplified case the problem is formally "solved" by Equation 2.9, practically we are nowhere near a solution. The joint probabilities $P(X, M)$ and $P(X, \overline{M})$ are unknown. In addition, if we were talking about probability density functions, *maximizing* $f_M(M|X)$ has little meaning. Nonetheless, previous researchers have explored the discrete case using maximum likelihood analysis. Since $\frac{P(M)}{P(\overline{M})}$ is a constant, the ratio to be maximized can also be expressed as $\frac{P(X|M)}{P(X|\overline{M})}$. The relationships between various image statistics and this ratio (termed the likelihood ratio) are explored in depth by Wesseley in [Wesseley76]. He reasons that given noise-free images,

$$P(X|M) = \delta(X - Y) \tag{2.10}$$

where

$$\delta(x) = \begin{cases} 1 & \text{if } x = 0 \\ 0 & \text{otherwise} \end{cases}$$

and

$$\delta(X) = \delta(x_1)\delta(x_2)...\delta(x_N)$$

Assuming no noise and perfectly matching regions is, of course, not realistic. Both images will be subject to noise, various geometric distortions and intensity offsets. Wesseley further assumes that all of these effects are additive, that they can be lumped under a single normal probability distribution $N(\mu_n, \sigma_n)$ with zero mean ($\mu_n = 0$), and that $N(n)$ and $f_X(X)$ are independent. Then, X, $(x_1, .., x_i, .., x_N)$ will be replaced by X_n, $(x_1 + n_1, ..., x_i + n_i, .., x_N + n_N)$. If X and Y are two element vectors, (x_1, x_2) and (y_1, y_2), then one can proceed to model the probability with noise as [Papoulis65]:

$$P(X|M) = \delta(x_1 - y_1)\delta(x_2 - y_2) * N(n_1)N(n_2) \tag{2.11}$$

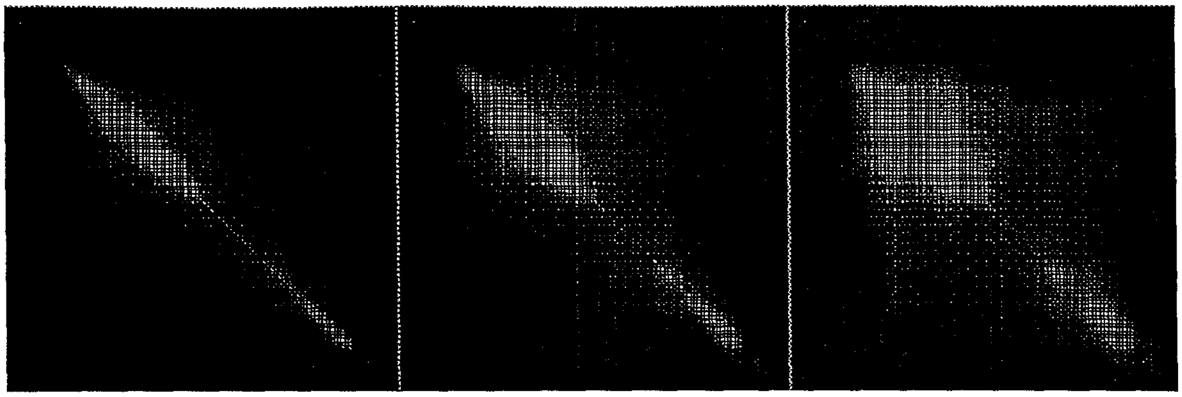


Figure 2.5: Joint probability distributions in a natural image obtained by pairwise pixel comparison.

The above plots are the results of an experiment carried out to demonstrate the coefficient of local spatial corelation in an image. Each pixel in the image is pairwise compared to pixels in its immediate neighborhood. Instance of a pair appears on the plot as a gray level (the higher the number of instances, the lighter the dot). We observe that pixel pair comparisons lead to ellipsoids around the $x = y$ axis, showing that there is significant correlation between pixels in a neighborhood. As the neighborhood expands, the ellipsoids become circular indicating that correlation decreases.

where $*$ denotes a convolution. The perfect match thus degenerates into the following:

$$P(X|M) = \frac{1}{2\pi\sigma_n^2} e^{\frac{-((x_1-y_1)^2+(x_2-y_2)^2)}{2\sigma_n^2}}. \tag{2.12}$$

The next task is to determine $P(X|\overline{M})$. For this analysis, I will assume that the components of vector X have pairwise jointly normal distributions:

Figure 2.5 was obtained from the same natural image (the histogram of which is shown in Figure 2.3), by pairwise comparison of pixel intensity values, (x_1, x_2), inside a one dimensional region. The highlighted regions show the incidence of pixel pairs (x_1, x_2). These form ellipsoids around the $x_1 = x_2$ axis, suggesting strong correlation between x_1 and x_2. From left to right, the regions are ±5, ±10, and ±20 pixels. The smaller the region, the higher the correlation coefficient.

If, for simplicity, we assume that distributions $f_{x_1}(\alpha)$ and $f_{x_2}(\beta)$ have zero mean and equal variance (i.e., $\mu_{x_1} = \mu_{x_2} = 0$ and $\sigma_{x1} = \sigma_{x2} = \sigma_i$), then

$$\rho_{x_1x_2} = \frac{E[x_1x_2]}{\sigma_i^2} \tag{2.13}$$

(The above equation suggests the inverse relationship between σ_i^2 and $\rho_{x_1x_2}$.) where

$$E[x_1x_2] = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \alpha\,\beta\, f_{x_1x_2}(\alpha,\beta)\,d\alpha\,d\beta \tag{2.14}$$

and

$$f_{x_1x_2}(x_1,x_2) = \frac{1}{2\pi\sigma_i^2\sqrt{1-\rho_{x_1x_2}^2}}\,e^{\frac{-1}{2\sigma_i^2(1-\rho_{x_1x_2}^2)}(x_1^2-2\rho_{x_1x_2}x_1x_2+x_2^2)} \tag{2.15}$$

Assuming joint normal distributions for x_1, x_2 which remain undisturbed given a no match, setting $\rho_{x_1x_2} = \rho_i$, and taking noise into consideration we have

$$P(X|\overline{M}) = \frac{1}{2\pi\sigma_{in}^2\sqrt{1-\rho_{in}^2}}\,e^{\frac{-1}{2\sigma_{in}^2(1-\rho_{in}^2)}(x_1^2-2\rho_{in}x_1x_2+x_2^2)} \tag{2.16}$$

where:

$$\sigma_{in}^2 = \sigma_i^2 + \sigma_n^2 \tag{2.17}$$

and

$$\rho_{in} = \frac{\sigma_i^2}{\sigma_{in}^2}\rho_i \tag{2.18}$$

The event M in Equations 2.10, 2.11, 2.12, 2.16 refers to M_{actual}. Note that assuming $P(X) \simeq P(X|\overline{M})$ loses its validity as the matches get less and less ambiguous.

The above analysis yields that the logarithm of the likelihood ratio, $L(X,Y)$, is proportional to a quadratic equation [Wesseley76]:

$$L(X,Y) \;\propto\; -\sigma_{in}^2(1-\rho_{in}^2)((x_1-y_1)^2+(x_2-y_2)^2)+\sigma_n^2(x_1^2-2\rho_{in}x_1x_2+x_2^2) \tag{2.19}$$

This equation defines a conic section on the (x_1,x_2) plane, rotated 45^o with respect to the axes (x_1,x_2). Furthermore, assuming that in practice ρ_{in} is always between 0 and 1:

$$L(X,Y) = \begin{cases} \textbf{an ellipse} & \text{if } \sigma_i^2 > \frac{\rho_{in}}{1-\rho_{in}}\sigma_n^2 \\ \textbf{a hyperbola} & \text{if } \sigma_i^2 < \frac{\rho_{in}}{1-\rho_{in}}\sigma_n^2 \\ \textbf{a parabola} & \text{otherwise} \end{cases}$$

The first part of the equation is essentially a similarity function, determining how close Y is to X. When the signal to noise ratio is high, or ($\frac{\sigma_n}{\sigma_{in}} \to 0$), this part dominates.

If the correlation coefficient is zero, i.e., $\rho_{in} = 0$, we have the equation for a circle where the center is dispaced from (y_1, y_2) by a multiplicative factor, $\frac{\sigma_{in}^2}{\sigma_i^2}$.

$$L(X,Y) \propto -(x_1 - (\frac{\sigma_{in}^2}{\sigma_i^2})y_1)^2 - (x_2 - (\frac{\sigma_{in}^2}{\sigma_i^2})y_2)^2 \tag{2.20}$$

This is close to the sum of squared differences similarity metric for image matching.

On the other hand, as the variance of the image and the coefficient of correlation tend to zero, the logarithm of the likelihood ratio becomes:

$$L(X,Y) \propto x_1 y_1 + x_2 y_2 \tag{2.21}$$

In N dimensions:

$$L(X,Y) \propto \sum_i x_i y_i \tag{2.22}$$

This is precisely the classical correlator, or the unnormalized product metric.

Thus, if the correlation coefficient is small, we have two appropriate similarity measures, depending on image statistics.

If we assume that the correlation coefficient is high, i.e., $\rho_{in} \simeq 1$, we face:

$$L(X,Y) \propto (x_1 - x_2)^2 \tag{2.23}$$

which is a crude measure of the variance of the image. This tells us that if the correlation coefficient is high, our best bet is to maximize image variance. Algorithms which only process regions where $\sigma_{in} >> \sigma_n$ by combing the image first with an *interest operator* to select the best regions to match exploit a related principle [BB82].

While the above analysis is very helpful for crystalizing some of the issues related to image matching, it fails to address three important points:

1. Although noise analysis was restricted to X, Y is also subject to noise.

2. We know in general that the coefficient of correlation ρ_i is **not** negligable. Therefore, a similarity metric alone can not maximize $P(M_{actual}|X)$ and thus can not be sufficient to provide a complete statistical solution to the image matching problem.

3. Because only $P(M_{actual}|X)$ is considered, ambiguity remains unaddressed.

I will look at this problem a little differently. I will consider $P(M_{possible}|X,Y)$ (as opposed to $P(M_{actual}|X)$) and assess the ambiguity of similarity metrics with noisy X **and** noisy Y:

As before, I will start by assuming that the two images we are trying to match are free of noise and only contain perfectly matching regions. Then all pixel by pixel actual matches must satisfy the condition $x = y$. To be considered a *possible* match, it is sufficient to satisfy the same condition, which makes the *a posteriori* probability a delta function:

$$P(M|X,Y) = \delta(Y - X) \tag{2.24}$$

Correspondingly, we might consider a probability distribution function to replace $P(M|X,Y)$ which equals the same delta function:

$$f_M(M|X,Y) = \delta(Y - X) \tag{2.25}$$

By similar reasoning used to obtain Equation 2.12, with noisy X,Y we have

$$f_M(M|X,Y) = P(M|X,Y) = \frac{1}{2\sigma_n\sqrt{2\pi}}\, e^{\frac{-(X-Y)^2}{4\sigma_n^2}} \tag{2.26}$$

Figure 2.6 shows this as a function of x and y. Note that this is still a number, not a probability distribution, and thus the evaluation

$$\max_i f_M(M|X,Y_i) \tag{2.27}$$

does yield a meaningful solution. Recall the initial objective of finding the Y_i that maximizes $P(M|X,Y)$. We can see that maximizing the logarithm of $P(M|X,Y)$ (or minimizing minus the logarithm) accomplishes the same purpose. So we can simplify the computation by replacing the function $P(M|X,Y)$ by another $M(X,Y)$ which is proportional to minus its logarithm and minimizing it.

$$M(X,Y) = (X - Y)^2 = \sum_i (x_i - y_i)^2 = \sum_i m(x_i, y_i) \tag{2.28}$$

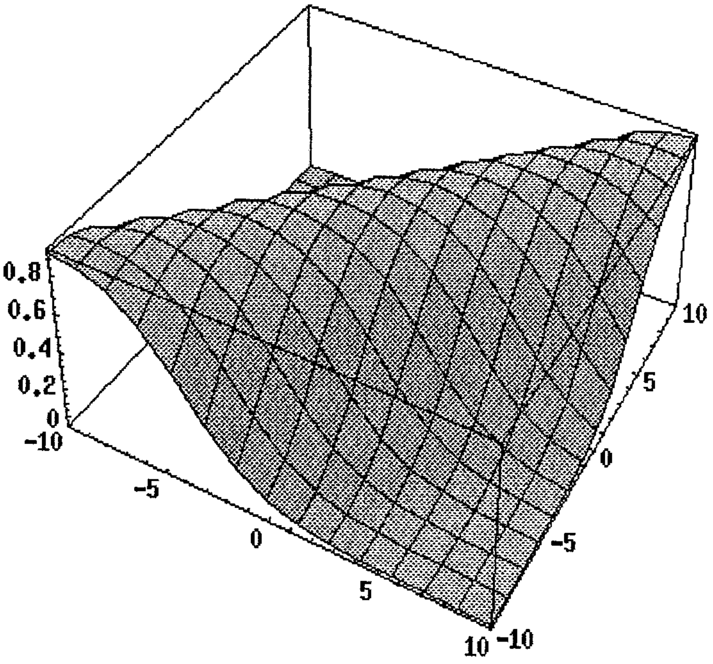


Figure 2.6: Gaussian function for $(x - y)$

Hence we end up with the sum of squared differences metric. Previously, this metric was shown to be most appropriate when the signal to noise ratio is high, or ($\frac{\sigma_n}{\sigma_{in}} \rightarrow 0$), and the correlation coefficient is small (i.e., $\rho_{in} \simeq 0$). Figure 2.11 shows values of $m(x, y)$.

The list of possible $m(x, y)$ are not exhausted by any means. I will examine several other such functions, or **similarity measures**, in Section 2.3.2.

To gauge the level of ambiguity in choosing among the possible matches, one can look at the probability distribution of m, $f_m(m)$. This is a function of the joint probability distribution $f_{xy}(x, y)$. The less the variance in $f_m(m)$, the greater the ambiguity of the match.

If we are conducting the matching process in real-time, we will know little or nothing about the global image statistics. In addition, the regions of the image will have their own distribution which may or may not be in tune with the global statistics. Consequently, in this analysis I will be interested only in the local values of $f_{xy}(x, y)$ and $f_m(m|x, y)$, which we **can** determine in real-time.

Again I will proceed given that the distribution of choice in the literature for $f_x(x)$ and $f_y(y)$ is the normal distribution.

For real images, we know that x and y are **not** independent, just like the elements of vector X. Due to significant spatial correlations within a region, pixels in the other image (y_i) will closely resemble x. Again assuming identical normal distributions $f_x(x)$ and $f_y(y)$, with $\mu_x = \mu_y = 0$ and $\sigma_x = \sigma_y = \sigma_i$, we obtain the following:

$$\rho_{xy} = \frac{E[xy]}{\sigma_i^2} \tag{2.29}$$

This suggests an inverse relationship between σ_i^2 and ρ_{xy}, just as Equation 2.13 did between σ_i^2 and $\rho_{x_1 x_2}$.

$$E[xy] = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x\, y\, f_{xy}(x, y)\, dx\, dy \tag{2.30}$$

where

$$f_{xy}(x, y) = \frac{1}{2\pi\sigma_i^2\sqrt{1-\rho_{xy}^2}}\, e^{\frac{-1}{2\sigma_i^2(1-\rho_{xy}^2)}(x^2-2\rho_{xy}xy+y^2)} \tag{2.31}$$

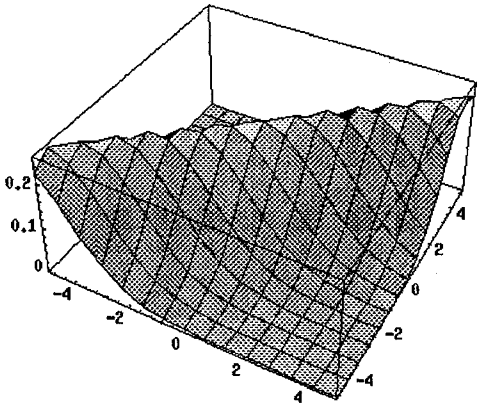


Figure 2.7: $f(y|x)$ when $\rho = 0.85$

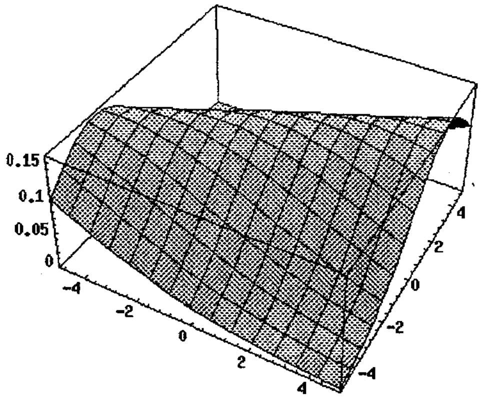


Figure 2.8: $f(y|x)$ when $\rho = 0.5$

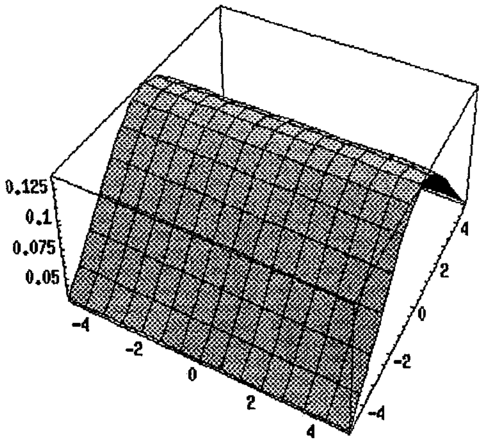


Figure 2.9: $f(y|x)$ when $\rho = 0$

We can also determine the conditional probability distribution $f_y(y|\mathrm{x} = x)$.

$$f_y(y|\mathrm{x} = x) = \frac{1}{\sigma_i\sqrt{2\pi(1-\rho_{xy}^2)}}\, e^{\frac{-1}{2\sigma_i^2(1-\rho_{xy}^2)}(y-\rho_{xy}x)^2} \tag{2.32}$$

We observe that

$$E[y|x] = \rho_{xy}x \tag{2.33}$$

The difficulty of the matching problem which we saw in the example is beginning to materialize once again in mathematical terms: For significant values of ρ_{xy} (near 1), $f_y(y|\mathrm{x} = x)$ assumes its highest values around the axis $(x = y)$, namely at points where x and y are most similar. The inputs to the similarity function $m(x, y)$ are clustered around $(x = y)$, making the determination of a clear maximum or minimum difficult. If images were prefiltered by a Gaussian kernel to reduce the effects of noise, this will increase ρ_{xy} and certainly worsen the ambiguity problem. Figures 2.7, 2.8, and 2.9 show $f_y(y|\mathrm{x} = x)$ for various values of ρ_{xy}. If no correlation exists between the two images, then we have the least ambiguous situation $f_y(y|\mathrm{x} = x) = f_y(y)$. Low values of σ_i^2 are also associated with high ambiguity.

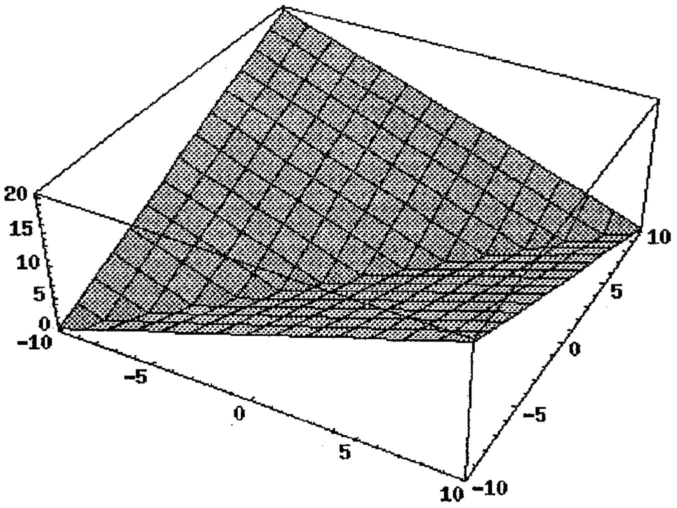


Figure 2.10: Absolute difference function

2.3.2 Several possible similarity measures

Here I will explore several popular similarity measures in terms of the probability distributions of their values. The probability distribution for (x, y) is taken to be that in Equation 2.31.

Absolute difference

$$m(x,y) = |x - y| \tag{2.34}$$

The analytical solution for $f_m(m)$ is easily obtainable:

$$f_m(m) = \frac{U(m)}{\sigma_i\sqrt{\pi(1-\rho)}}\, e^{-\left(\frac{m^2}{4\sigma_i^2(1-\rho)}\right)} \tag{2.35}$$

Squared differences

$$m(x,y) = (x - y)^2 \tag{2.36}$$

The probability distribution $f_m(m)$ is

$$f_m(m) = \frac{U(m)}{2\sigma_i\sqrt{\pi m(1-\rho)}}\, e^{-\left(\frac{m}{4\sigma_i^2(1-\rho)}\right)} \tag{2.37}$$

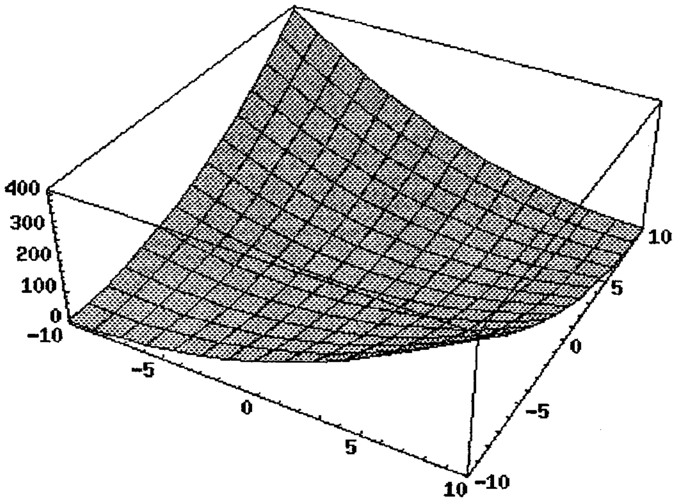


Figure 2.11: Squared difference function

Analytical solutions of the probability distribution $f(m)$ for the difference and difference squared metrics are shown in Figure 2.12. For both metrics, $f_m(m)$ is high for low values of m. Both metrics need to be minimized to obtain the correct disparity. Consequently, comparisons are likely to contain significant ambiguity when using these metrics, especially the difference squared metric. Experimental values obtained from the rock image (Figure 3.16) are shown in Figure 2.13. The analytical and experimental distributions compare rather favorably. Experimentally, though, we observe:

1. An offset between the two images, most likely due to an average difference in illumination between the two images.

2. A higher variance in the Gaussian distribution, which most likely means that a parameter adjustment is needed in plotting the analytically obtained function(s).

3. Singularities in the distribution, a byproduct of the digitatization process for the photograph. We can observe similar singularities in the top plot of image intensity distributions in Figure 2.3.

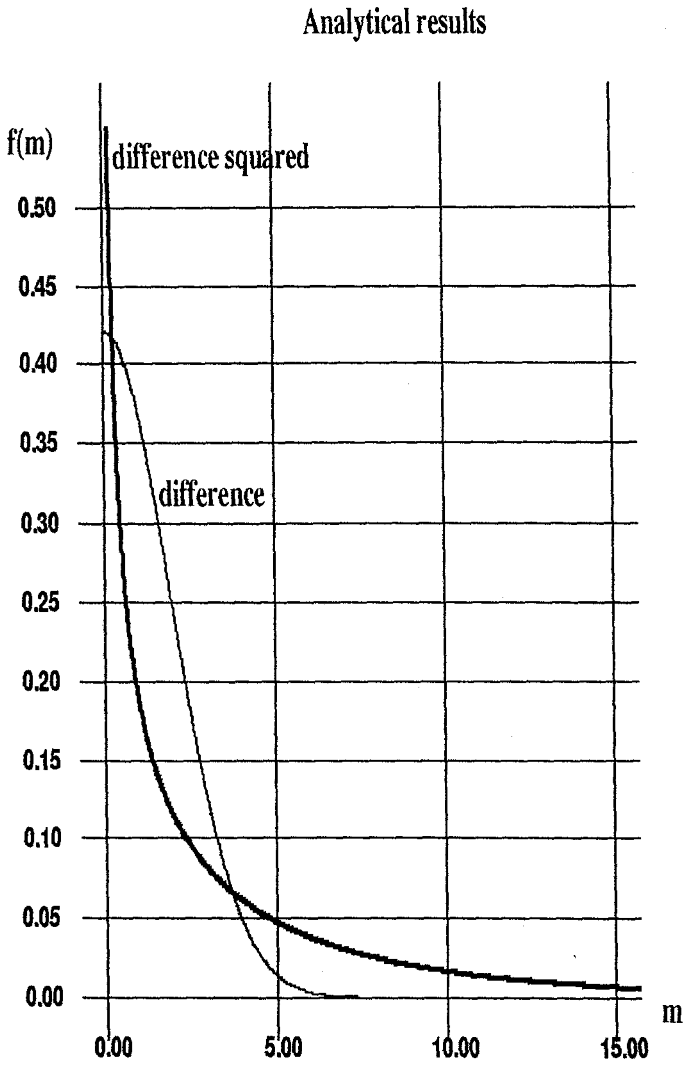


Figure 2.12: $f_m(m)$ obtained analytically for the difference and the difference squared metrics

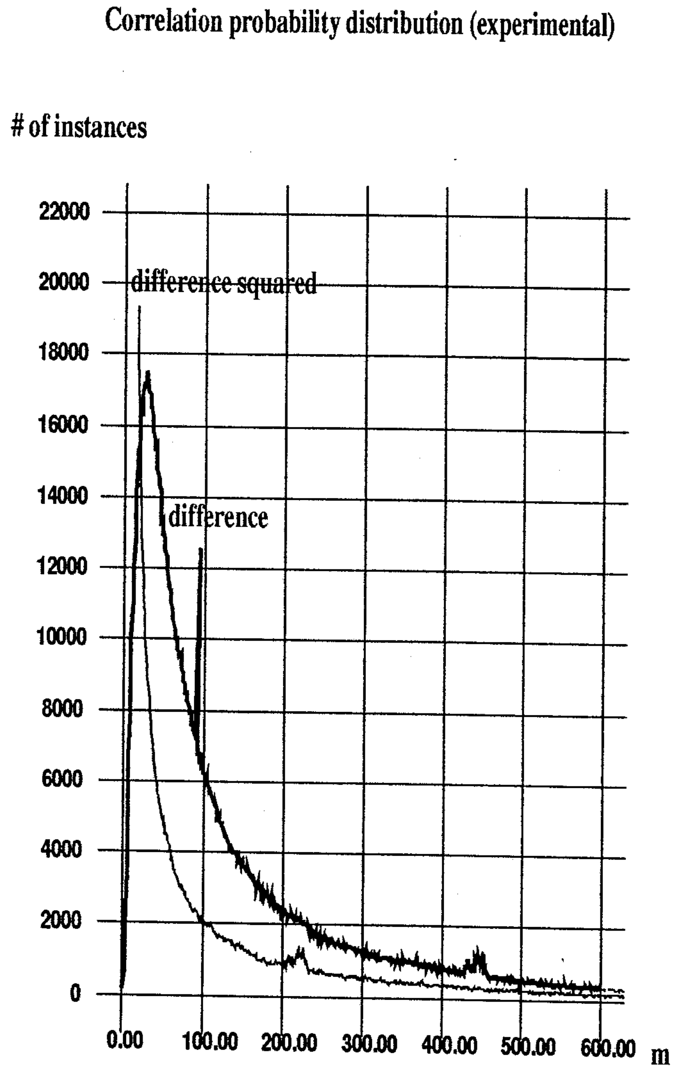


Figure 2.13: Experimental probability distributions for the difference and the difference squared metrics

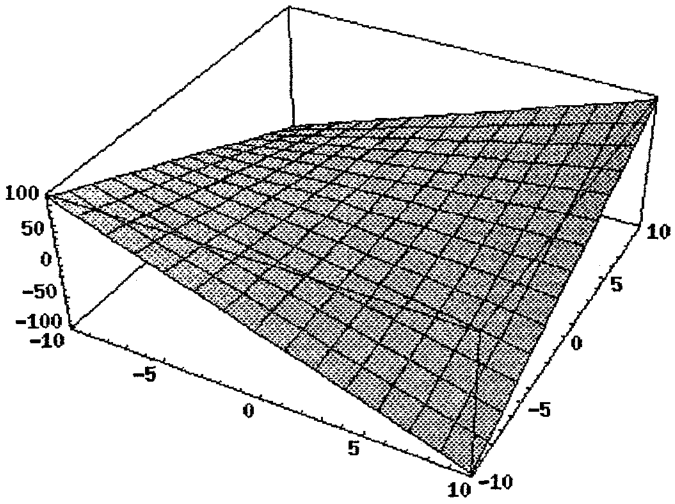


Figure 2.14: Inner product (correlation) function

Inner product, or correlation

$$m(x,y) = xy \tag{2.38}$$

Normalized inner product

$$m(x,y) = \frac{xy}{\sqrt{x^2+y^2}} \tag{2.39}$$

Analytical solutions for the inner product and normalized inner product metrics are not as easily obtainable. Experimental values obtained from the rock image (Figure 3.16) are shown in Figure 2.16.

A hardware metric alternative

$$m(x,y) = \frac{1}{1+\frac{4}{w}cosh^2(x-y)} \tag{2.40}$$

The parameter w is adjustable and can be utilized to contain a parameter similar the σ_n^2 in Equation 2.26. The Figure 2.17 shows a Gaussian-like distribution around the central

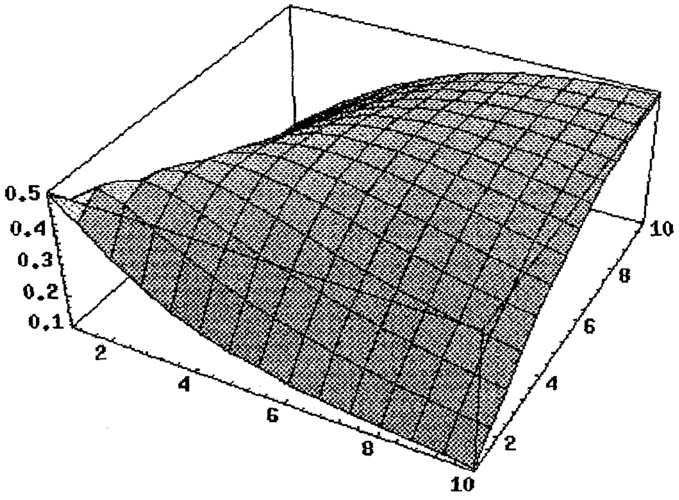


Figure 2.15: Normalized inner product function

line of $x = y$. The graph is very similar to the plot in Figure 2.6.

This is a well-behaved function:

1. Its integral is finite unlike the integrals of the other metrics.

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} m(x,y)\,dx\,dy \quad . \tag{2.41}$$

2. Within arbitrary range of (x, y), it can be bound above and below by a scaled Gaussian function, the peak of which coincides with that of the metric at $x = y$.

Probability distribution $f_m(m)$ for the hardware metric was obtained experimentally, again using the rock image (Figure 2.18).

2.3.3 Discussion

Computation which uses the properties of a physical medium is bound to have limited precision.

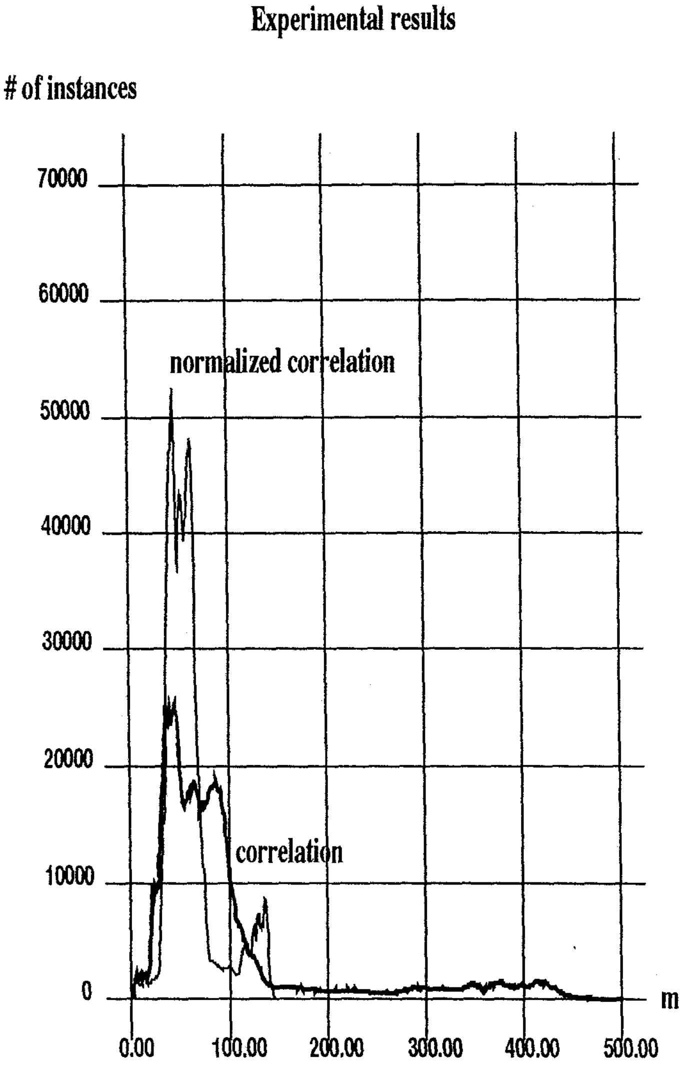


Figure 2.16: Probability distributions for the correlation and the normalized correlation metrics

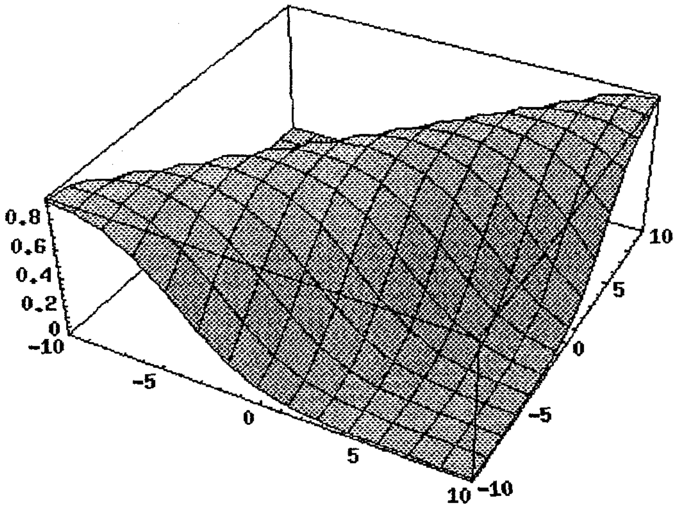


Figure 2.17: Hardware metric function

In computer simulations, computation can be carried out with high precision numbers. Consequently, the similarity metric values can be evaluated much more accurately than they would be using an analog VLSI chip. Thus, there will be many instances where a computer program will determine the correct maximum of the metric outputs and an analog chip will not. The probability of such instances rises if the variance of the metric distribution is low. In other words, if the metric values are clustered around a single value (as in the case with the difference squared metric), determination of a maximum is more difficult and requires higher accuracy. On the other hand, if the metric values are distributed uniformly across a range (as in the case with the hardware metric), ambiguity introduced by the metric is less and determining the maxima of metric values is less likely to be challenging.

This is precisely the reason why the probability distribution of a metric that has limited accuracy due to the physical implementation is an important issue.

Please note that in the experimental plots the actual metric values have been scaled to

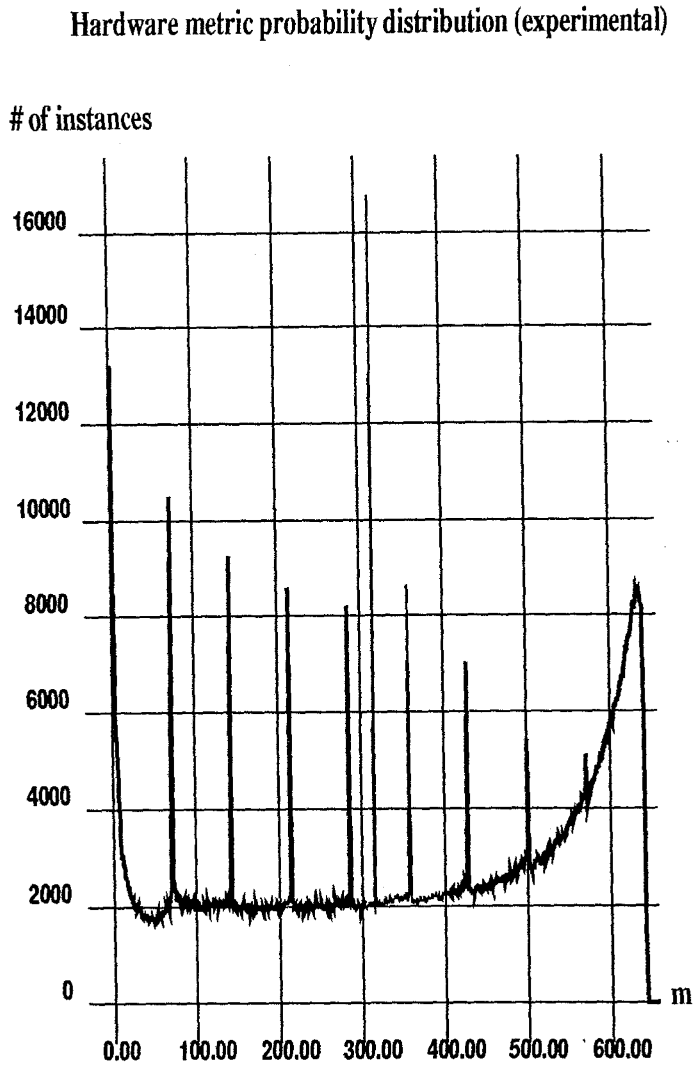


Figure 2.18: Probability distribution $f_m(m)$ for the hardware metric

The distribution is almost uniform in a range of values, leading us to conclude that it has a higher variance than the other metrics when scaled to cover the same range. The singularities arise from the singularities in the image itself (see Figure 2.3) and possibly from the nature and limits of the numerical computation.

cover roughly the same range. Only then we can compare them to evaluate their precision requirement in a physical computation medium.

Experimental probability distributions computed in this manner show that the hardware metric is indeed the metric with the most favorable (i.e., the highest) variance. Therefore, we expect the comparisons made using this metric to lead to the least amount of ambiguity.

The metric has a clear upper bound for all (x, y), which is another favorable feature.

The following chapters will explore the simulation and implementation of a stereo correspondence algorithm using this metric.

Chapter 3
Simulating the Hardware Matching Algorithm

3.1 Describing the algorithm

The algorithm I propose is essentially an area-based correlation scheme. There are two parts to the algorithm:

1. Solving the correspondence problem;

2. Assigning a confidence to that solution.

3.1.1 Solving the correspondence problem

Image matching is carried out between the stereo pairs, exactly as described in Section 2.3.1 in the previous chapter: The region selected in one image is compared with candidate regions in the other image and exactly one region is selected as its match.

Prior to processing, the images are filtered by an *exponential* filter to reduce the undesirable effects of noise. As previously mentioned, Gaussian filters increase the coefficient of correlation between the pixels of the stereo pair and increase ambiguity. The exponential filter is generally less prone to such an effect because its kernel puts less weight on immediate neighbors than the Gaussian kernel.

Filtered pixel values are used to compare neighborhoods in each image. A comparison of the two filtered neighborhoods is made at each possible disparity value. The corresponding region is identified from among the candidate regions utilizing the hardware metric mentioned in the previous chapter (Section 2.3.2). Assuming that the neighborhood is two dimensional, with width $2\kappa + 1$ and height $2\lambda + 1$, the value of the matching function at image coordinates (x, y), for a given horizontal disparity δ_x $(M(x, y, \delta_x))$, is:

$$M(x, y, \delta_x) = \sum_{j=y-\lambda}^{j=y+\lambda} \sum_{i=x-\kappa}^{i=x+\kappa} \frac{1}{1 + \frac{4}{w} cosh^2(\frac{\kappa}{2kT}(I_R(i,j) - I_L(i - \delta_x, j)))} \tag{3.1}$$

Where w and κ are hardware circuit parameters, kT is a constant, δ_x is the disparity, and $I_R(x,y)$ and $I_L(x,y)$ are filtered pixel values of the right and the left image respectively. The region that generates the highest comparison sum is identified as the corresponding region. Assuming that the allowed disparity range is between $-\Delta$ and Δ, this can be written as:

$$Disparity(x,y) = \delta_x \quad : M(x,y,\delta_x) \quad = \max_{-\Delta \leq \xi \leq \Delta} M(x,y,\xi) \quad \leq \frac{w(2\kappa+1)(2\lambda+1)}{w+4} \tag{3.2}$$

The above inequality stems from the bounded nature of the hardware metric. Unlike many other metrics mentioned in the previous chapter, for any value of I_R and I_L, the metric always stays below a known maximum.

3.1.2 Setting the confidence value in the solution

As discussed in the previous chapter, ambiguity prohibits the hardware metric (or any other single similarity metric) from solving the image matching problem. To remedy this situation, a confidence measure is introduced.

In the previous chapter I explained about the concept of an interest operator that combs the images to determine high variance regions to reduce the ambiguity problem. We know that we can think of the metric as a function of the image. Variance of a function $g(x)$ can be approximated by [Papoulis65]

$$\sigma^2_{g(x)} \simeq (g'(x))^2 \sigma^2_x \tag{3.3}$$

Thus, instead of identifying the regions of an image where the image standard deviation σ_i is high, one can use the confidence measure to identify regions where the deviation in the value of the metric (σ_m) is high. These two regions will coincide if the function g is well behaved. By definition,

$$\sigma^2 = E[(x-\eta)^2] \tag{3.4}$$

where η is the mean value of x. We can write this also as

$$\sigma^2_m = E[(m)^2] - \eta^2_m \tag{3.5}$$

Figure 3.1 shows an identical stereo pair composed by step edges subjected to the variance computation. The locations of the step edges are circled. The graph shows that the variance values do exhibit an "M" pattern around the step edge, where the maximum metric

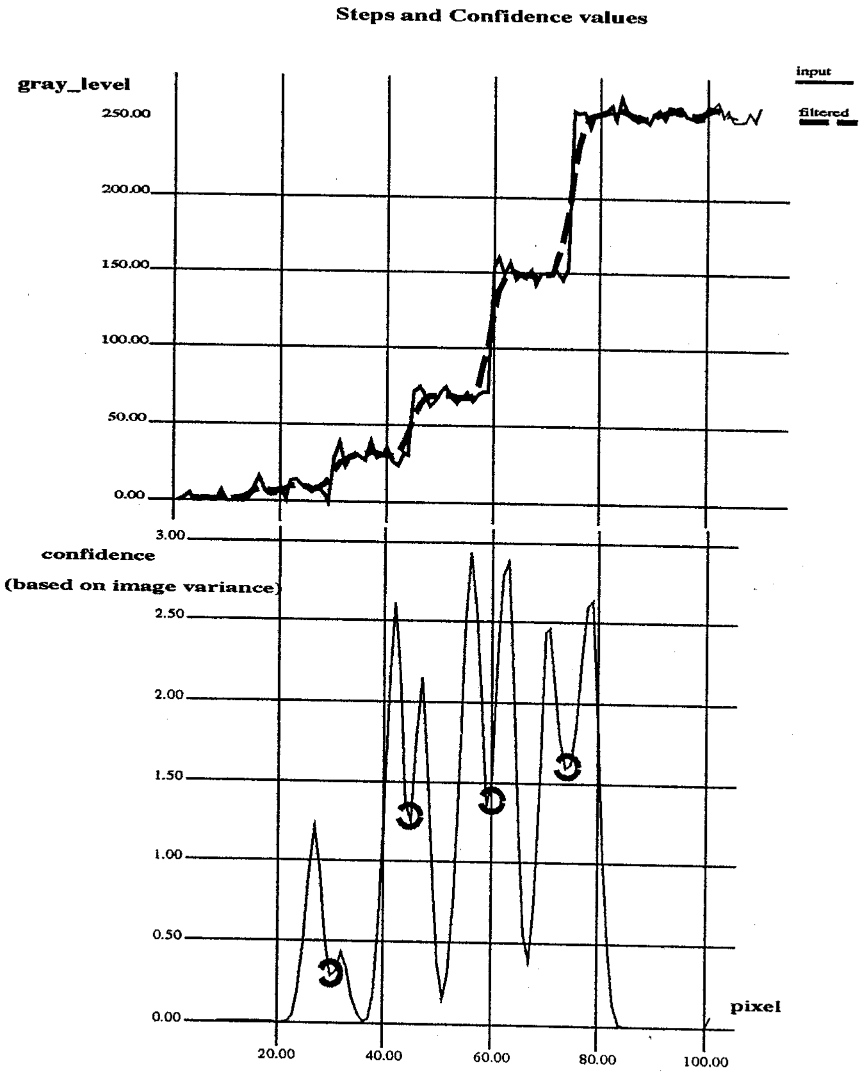


Figure 3.1: Steps: Confidence calculated by the variance method

The confidence metric here is equal to the variance of the metric values for each pixel. (The allowed range of disparities is ± 5 pixels, and consequently there are 11 metric values for each pixel.) Because of the smoothing, the metric variance at the exact location of the edge exhibits a local minima.

test yields the correct disparity and should be assigned high confidence. The local minima of the variance at the location of the edge is due to smoothing of the step edge. We observe that Equation 3.3 does not quite hold for this metric. In addition, the detection of the "M" pattern poses additional problems.

For these reasons, basing the confidence metric on the image variance is not a reliable approach, and a different method for assessing the ambiguity problem is called for. Figure 3.2 shows the values that the hardware metric acquires as a function of disparity, for various step edges and noise deviation, σ_n. The values of both are shown in the plots. (There are 256 gray levels.) Step size refers to the intensity difference between the sides of the step edge. We can extract confidence information from the shape of this curve. Thus, I diverge from the statistical analysis and instead evaluate the sharpness of the peak of the metric. Two possible methods for assessing the peak that are easy to compute are

1. <u>Derivative method:</u> The sharpness of the peak can be assessed from the first derivative. Figure 3.3 shows the confidence evaluated using this method for the step edges. We observe that the peaks coincide with the locations of the edges.

   $$Confidence(x,y) = M(x,y,\delta) - \max_{-\Delta\leq\xi\leq\Delta}^{(\xi\neq\delta)} M(x,y,\xi) \qquad (3.6)$$

   where

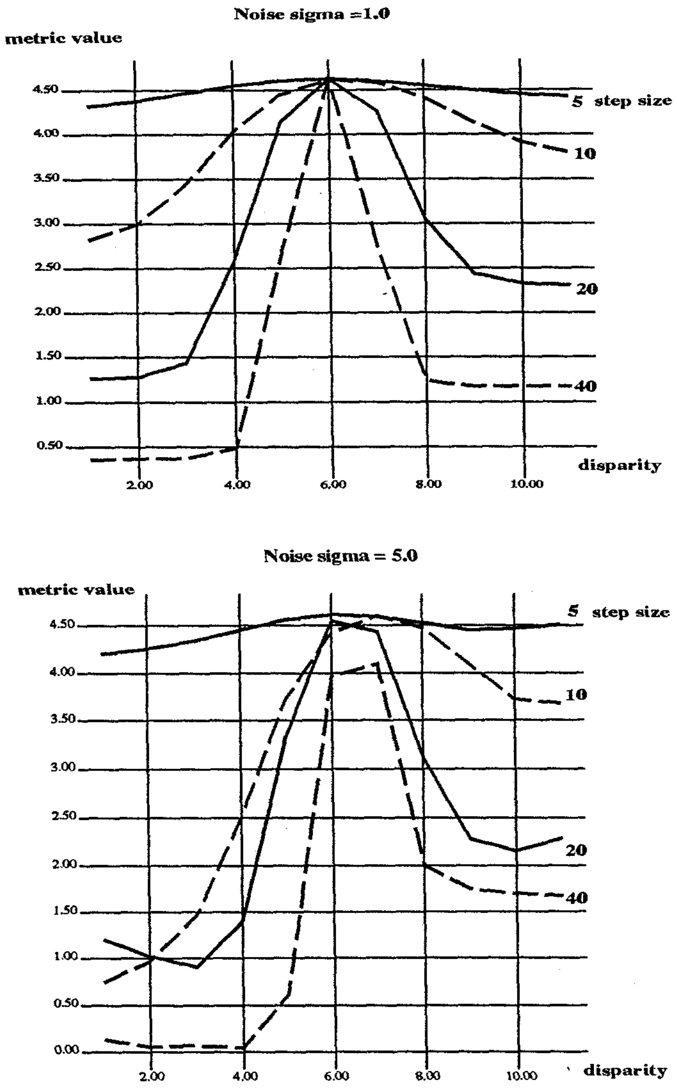
   $$M(x,y,\delta) = \max_{-\Delta\leq\xi\leq\Delta} M(x,y,\xi) \qquad (3.7)$$

2. <u>Ratio method:</u>

   $$Confidence(x,y) = \frac{M(x,y,\delta)}{\sum_{\xi=-\Delta}^{+\Delta} M(x,y,\xi)} \qquad (3.8)$$

   where $M(x,y,\delta)$ is as described in Equation 3.7. Figure 3.4 shows the confidence evaluated using this method for the step edges. We again observe that the peaks coincide with the locations of the edges.

A comparison between the confidence values calculated using the two methods and the image variance is illustrated in Figure 3.5. Variance was calculated inside a five-pixel window. We observe that although the three plots have different scales, their shapes are very compatible. I will show how these two metrics perform in application in the Section 3.2.2.


Noise sigma =1.0

Noise sigma = 5.0

Figure 3.2: Steps: Hardware metric evaluated for various σ_n and step sizes

There are eleven possible disparities and value 6 on the x-axis corresponds to zero disparity. Metric value distributions are shown at the exact location of the edge. Peaks get more pronounced as the step size, or the difference in intensity value on two sides of the edge, increases. Peak location becomes less reliable as noise variance is increased.

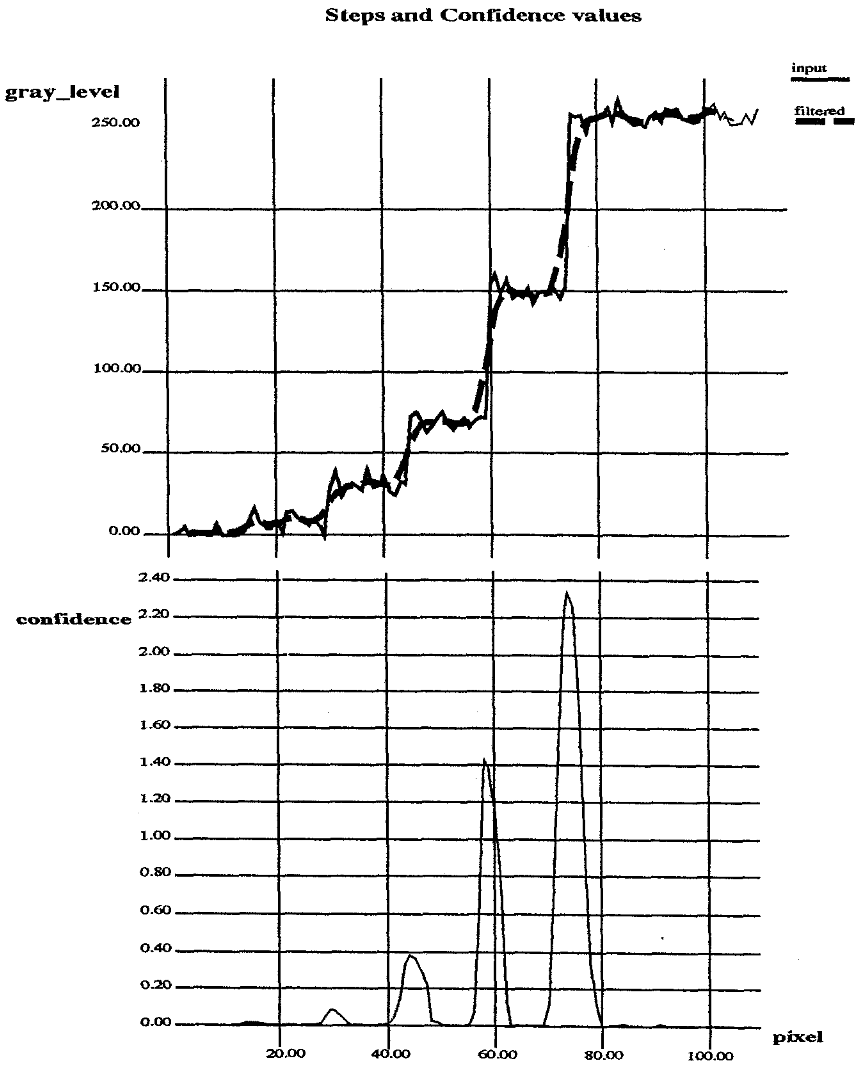


Figure 3.3: Steps: Confidence calculated by the derivative method

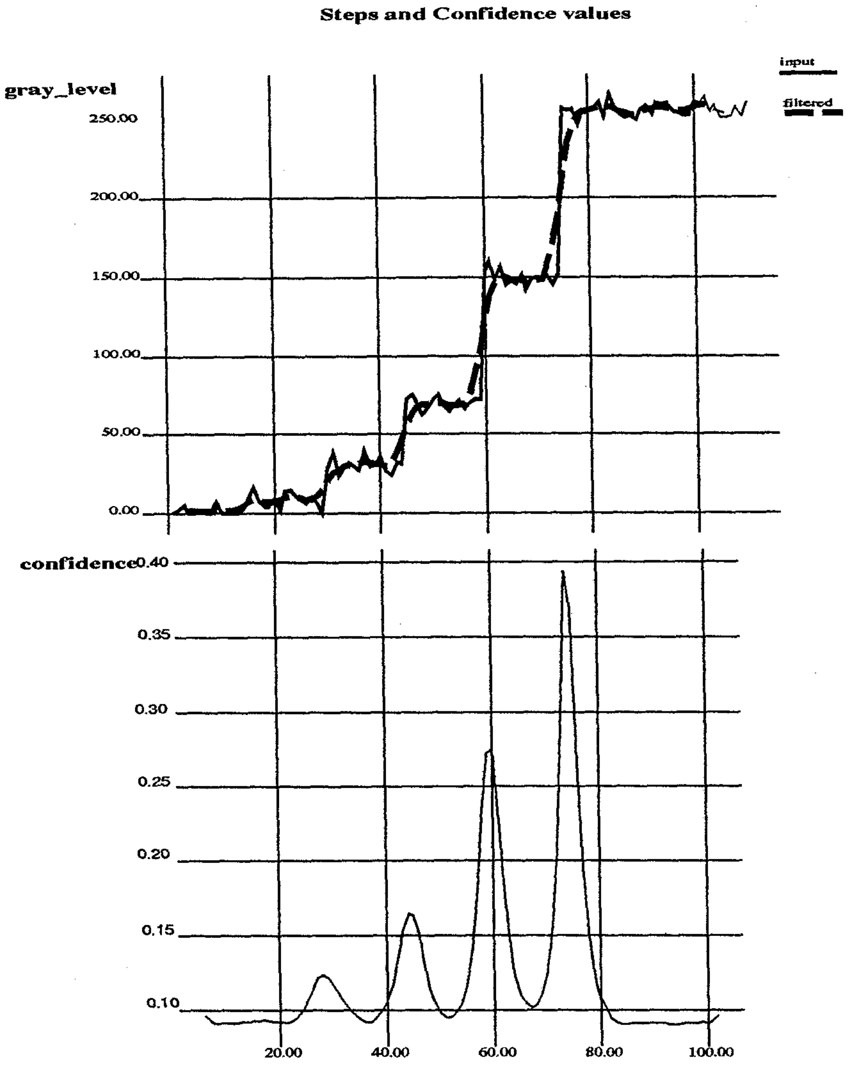


Figure 3.4: Steps: Confidence calculated by the ratio method

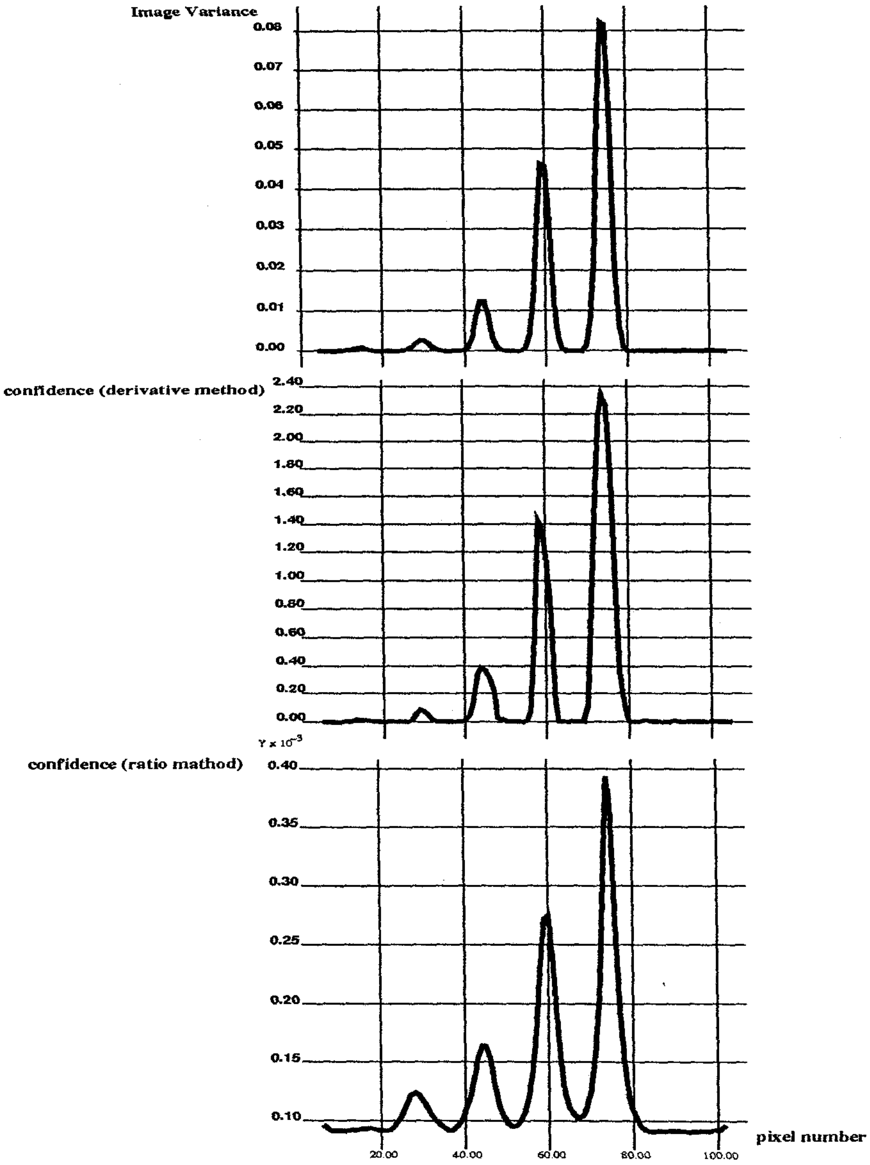


Figure 3.5: Confidence metric compared to image variance

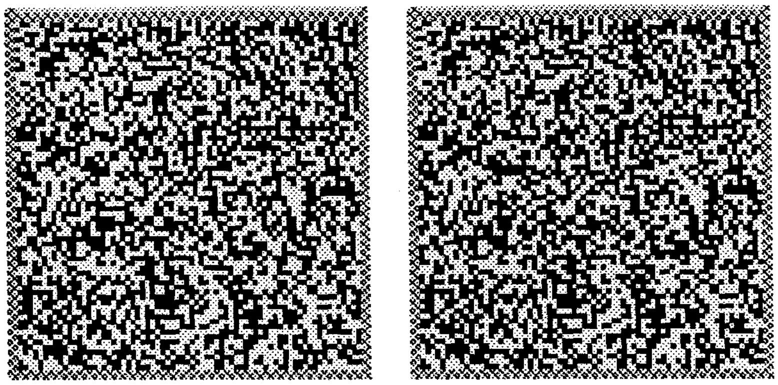


Figure 3.6: Random dot stereogram pair

3.2 Simulation results

3.2.1 Random dot stereograms (RDS)

Random dot stereograms are image pairs composed of various gray level pixels arranged in a random pattern (Figure 3.6). One of the images is usually a replica of the other, except for regions strategically displaced against those in the other image to create a sense of depth. When each image is presented to each eye, the observer gets the sensation of viewing surfaces at different depths because of these displacements, or disparities.

Target density

The typical binary RDS contains 50% white and 50% black dots or pixels. As one increases the percentage of white or black dots, target density decreases, leading to an increase in essentially featureless regions in the image. An RDS made up of all white or all black pixels contains no information for image matching. Decreased target density causes the image matching problem to become more ambiguous. I have carried out simulation and hardware experiments to study the effects of adjusting the percentage of black dots in an RDS. We could look at this as the effect of adjusting target density on image matching. Figure 3.7 shows RDS's with decreasing target density, from left to right, (50%, 30%, 20%, 10% and 5%). Figure 3.8 shows the simulation results in the same order from left to right. It is readily observable that decreasing the target density leads to degradation

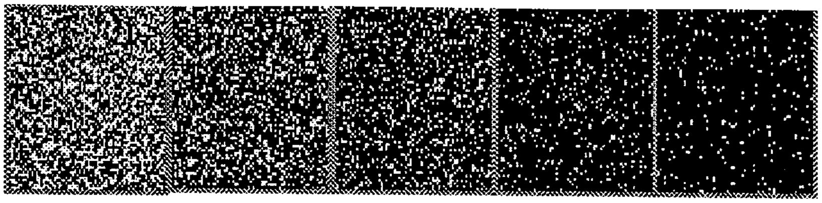


Figure 3.7: Adjusting target density in RDS's

From left to right target density values are 50%, 30%, 20%, 10% and 5%. Target density is adjusted by decreasing the probability of white pixels in the RDS image generation program.

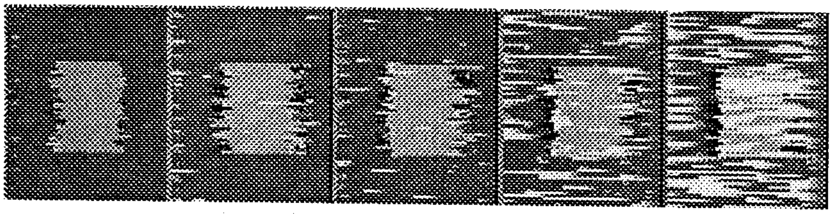


Figure 3.8: Simulating decreasing target density

of performance. Hardware test results are reported in the next chapter.

3.2.2 A synthesized image

This image pair (Figure 3.9) is courtesy of Prof. D.G. Jones of McGill University. Image was used to evaluate the performance of the stereo matching algorithm in the reference [JM92]. This is a synthesized image with interesting features. The background is similar to a gray-level random dot stereogram. The geometry is convergent with significant vertical disparity at the corners. Image pair contains many occlusion points, some of which extend over many pixels.

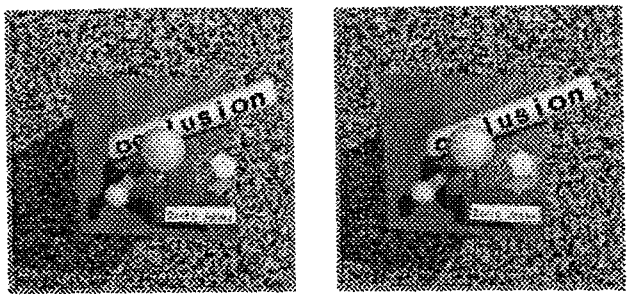


Figure 3.9: The synthesized image pair

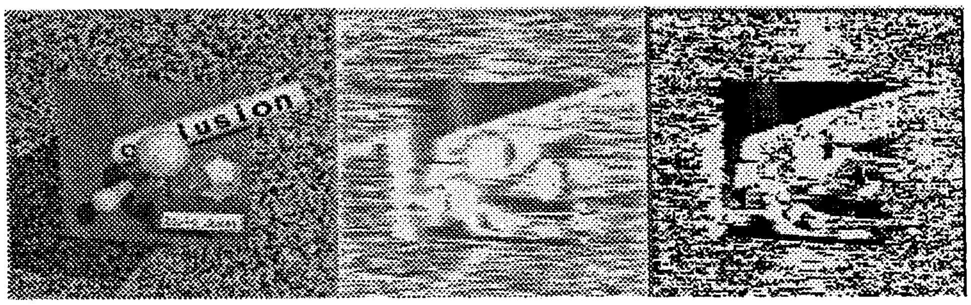


Figure 3.10: Confidence metrics using the ratio and derivative methods

Results with two different confidence metrics

I have described two confidence metrics, one obtained by the derivative method and the other by the ratio method. Simulation results in Figure 3.10 show their values with the synthesized image pair. The confidence values were based on the hardware metric and have been appropriately scaled to form the confidence maps.

Adjusting the confidence threshold

Figure 3.11 contains disparity maps interpolated between high confidence points only. The threshold that defines a "good" disparity point was lowered gradually from left to right. Lowering the confidence threshold leads to an improved, higher resolution disparity map which indicates that image is full of features and image matching is generally not as

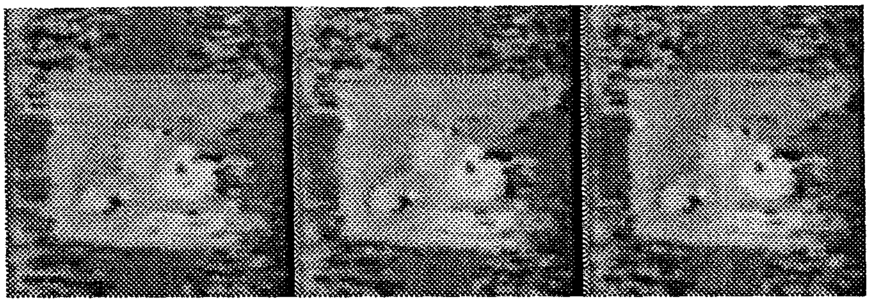


Figure 3.11: Adjusting the confidence threshold with the synthesized image

ambiguous as it is for many natural images.

Smoothing

The amount of smoothing in simulations can be increased by raising the variance of the exponential filter, σ^2_{filter}. Smoothing causes each matching window to contain information from pixels outside the window, leading to a more global matching framework. Beyond an ideal value of σ_{filter}, however, smoothing begins to introduce increasing ambiguity. I have carried out simulation and hardware experiments to demonstrate this. Figure 3.12 shows the simulation results. Hardware test results with an RDS are reported in the next chapter.

The five disparity maps in Figure 3.12 have been obtained with σ_{filter} values of 0.5,1.0,1.8, 2.8 and 5.8 pixels from left to right. No thresholding or interpolation was carried out. Error analysis on these images, comparing them to the two dimensional simulation results in Figure 3.15, confirmed that initially smoothing the stereo pair improves the disparity results: Results with $\sigma_{filter} = 1.0$ are better than those with $\sigma_{filter} = 0.5$. But beyond $\sigma_{filter} = 1.0$, smoothing seems to degrade the performance of the hardware metric.

Performance with respect to this error analysis is based on the variance of the error, σ^2_e. Higher σ^2_e signals degraded performance. Table 3.1 shows the ideal σ_{filter} to be around 1.0 pixels.

	smoothing factor : σ_{filter}				
	0.5	1.0	1.8	2.8	5.8
error average, η_e	0.14	0.15	0.14	0.13	0.09
error variance, σ_e^2	0.42	0.40	0.42	0.44	0.48

Table 3.1: Error in disparity from images smoothed by exponential filter of σ_{filter}

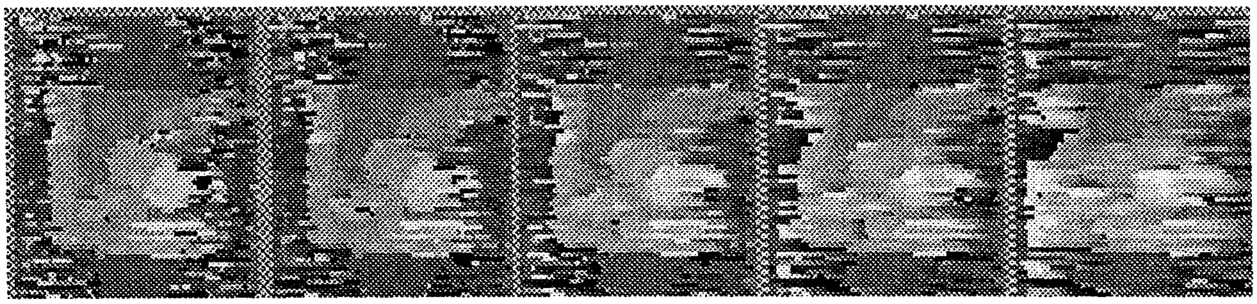


Figure 3.12: Smoothing degrades matching

The five disparity maps have been obtained with σ_{filter} values of 0.5,1.0,1.8, 2.8 and 5.8 pixels from left to right. No thresholding or interpolation was carried out.

Figure 3.13 and Figure 3.14 show the confidence values calculated using the derivative and ratio methods respectively, with the same diffusion lengths. As σ_{filter} increases, the areas with low confidence also increase. Over-smoothing degrades confidence performance as well as disparity performance of the algorithm.

Including the second dimension

Two dimensional image matching simulations were carried out with the hardware metric. Including the second dimension brings along three important improvements:

1. Disparity results are accurate in the corners of the image since vertical disparity is corrected for.
2. Image matching region expands from being a string of pixels to a two dimensional region of pixels. The correct match is identified based on a wider range of support

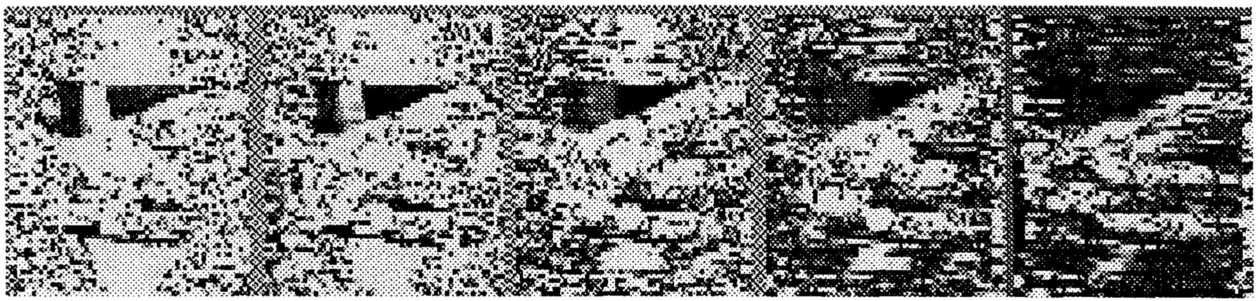


Figure 3.13: Smoothing degrades confidence values obtained by the derivative method

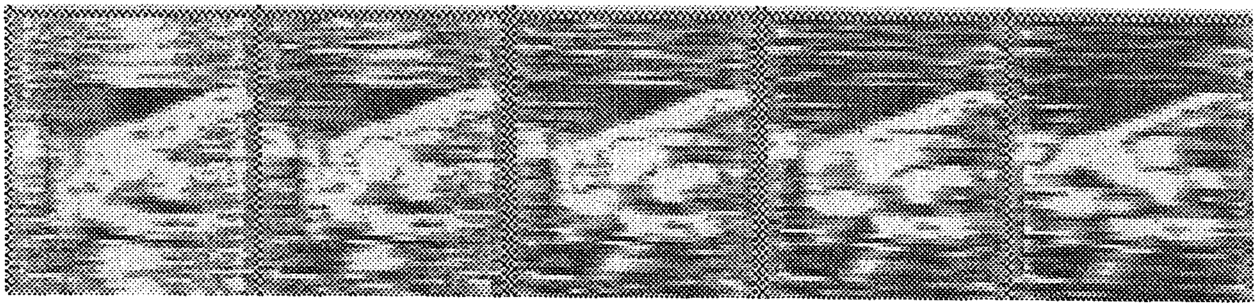


Figure 3.14: Smoothing degrades confidence values obtained by the ratio method

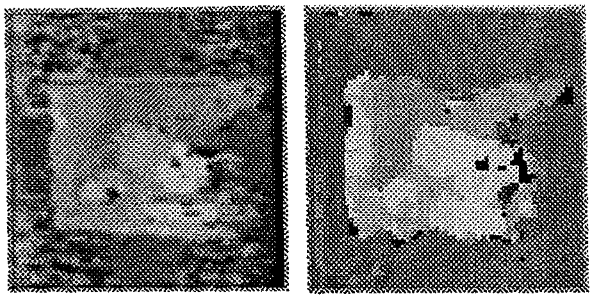


Figure 3.15: One dimensional and two dimensional simulations compared

and the results are generally more accurate.

3. Again with the aid of the second dimension, jagged disparity discontinuities are reduced.

As expected, two dimensional application of the algorithm takes significantly longer to simulate. Hardware implementation to be described in the next chapter is limited to one dimension.

3.2.3 Natural images

Rocks image

This image pair is courtesy of L. Matthies of Jet Propulsion Laboratories. This is a photograph of a scene outside JPL in Pasadena, California.

Figure 3.16 shows the image and the disparity map obtained with the hardware algorithm.

Train image

This image pair is also courtesy of L. Matthies. It is a photograph of the maquette of a small town scene. The image has been used for evaluating other algorithms [Szeliski89].

Figure 3.17 shows the image and the disparity map obtained with the hardware algorithm.

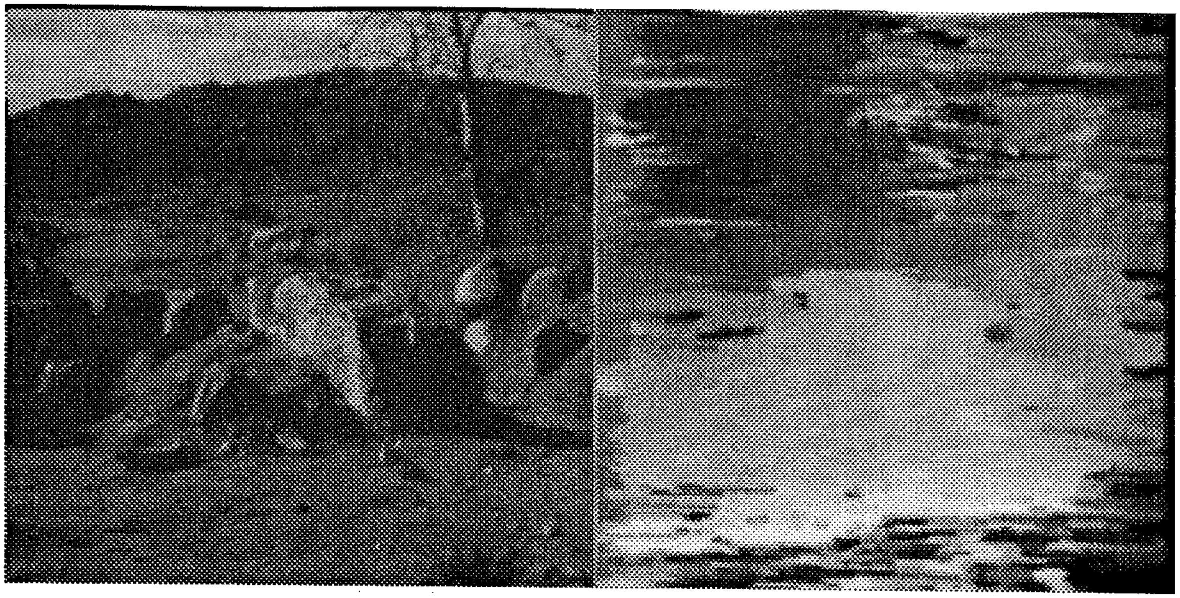


Figure 3.16: The rocks image and disparity map

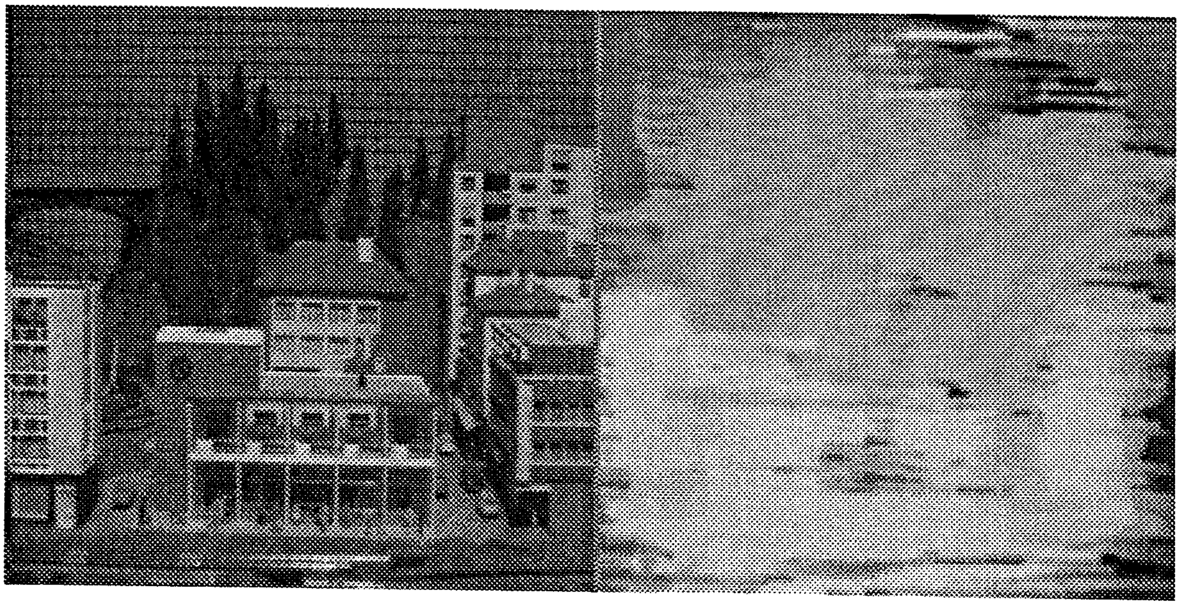


Figure 3.17: The train image and disparity map

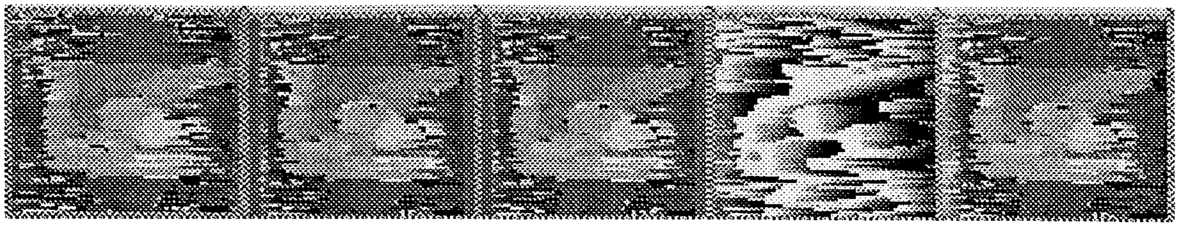


Figure 3.18: Disparity maps obtained using five different similarity metrics

Form left to right, the metrics utilized are hardware,difference, difference squared, cross correlation and normalized cross correlation metrics. The cross correlation metric has the worst performance. The rest are fairly compatible for this image.

3.3 Comparison with other similarity measures

In the previous chapter I have explored several similarity measures. Here, I will compare my hardware metric with them. The disparity maps in the Figure 3.18 were obtained using, from left to right, the hardware, difference, difference squared, cross correlation and normalized cross correlation metrics.

As one can see, the (unnormalized) cross correlation metric yields inaccurate results. The remaining metrics are quite compatible. Table 3.2 shows the average error and variance of the error. The reference disparity is the result of the two dimensional simulation (Figure 3.15). The hardware metric performance is compatible with that of popular metrics.

Similarity metric (in the order of performance)	η_e	σ_ϵ^2
Difference squared	-0.14	0.37
Difference	-0.14	0.38
Hardware metric	-0.15	0.39
Normalized correlation	-0.15	0.41
Correlation	-0.02	0.75

Table 3.2: Disparity error in images obtained by different similarity metrics

Chapter 4
Analog VLSI Implementation

4.1 Comparison to previous work

I mentioned in Section 1.2.6 that systolic array type chips have been used to solve various problems in vision.

There are various hardware applications specifically designed for stereopsis. Several recent ones are listed below:

1. Simulations of a CCD/CMOS implementation for a stereo vision system [HLLW91].
2. A resistive network analogy to a variational solution to the problem of depth from stereo [CG89].
3. Another depth finding method by locating light stripes projected on a scene: Prototype system which acquires and processes range data by light-stripe range finding [GCK91].

In Section 1.2.6, I also mentioned the recent analog VLSI approach to vision problems, especially that of Mahowald in solving stereo correspondence [Mahowald92]. This work follows the same tradition, but introduces a feedforward, serial approach to overcome the computational density and communication challenge of handling two dimensional images in real-time. I believe it is important to outline the similarities and differences between my approach and Mahowald's in detail. Differences lie in the following areas:

1. Elimination of the target selection process: As was outlined in Section 1.2.1, solving the stereo correspondence problem begins with the selection of targets to match in the two images. Mahowald begins with the assumption that the two images have been processed already and targets have been marked to produce a binary image (at each pixel a target is present or not).

Determining the locations of targets (or edges) in an image is a well-explored problem and many superb algorithms exist. Yet, it would be difficult to find a real-time hardware implementation that produces edges of good enough quality to match between two images. Extra or missing edges between images could easily throw the matching process off course.

By using intensity values from the two images without any preprocessing, this hardware implementation essentially eliminates the problematic stage of target selection from the solution.

2. Increased dynamic range of images: As mentioned, Mahowald uses a binary image for the matching stage. Binary images make the matching process more robust against false matches caused by unintended circuit behavior. Hardware test results show that this implementation performs well with images of 256 gray levels (voltage range 1.5 – 3.5 V) as well as with binary images (1.5 V is a black pixel, and 3.5 V is a white pixel) (Section 4.3).

3. Serial computation and windowing for 2-D images: Mahowald computes the solution to the matching problem with all the targets present simultaneously along an epipolar line. Although this allows for the use of a global algorithm to resolve ambiguities, it also limits the size of the image that can be processed. My implementation uses windowing to serialize the computation. It sacrifices the ability of global computation for the ability to process any size two dimensional image.

4. Feedforward computation and confidence measure: Global computation in the form of feedback from neighboring computing units or from other spatial scales are parts of the Mahowald stereo chip. These are powerful mechanisms for resolving the ambiguities that are part of the matching process.

   My implementation takes a different approach to resolving ambiguities: It leaves them unresolved. In processing real images this poses only a limited problem as hardware test reveals (Section 4.3). Ambiguities arise in several ways:

(a) Lack of features: Regions of the image where intensity values have a narrow distribution also lack targets. So matching such points without peripheral information is not possible. In the Mahowald chip, because all targets along the epipolar line are considered, one can safely assume that at least some targets will be visible and regions without targets will be assigned disparity values from the interpolation between matched targets. In my implementation, windows that lack features will output a disparity with low-confidence, signaling the next stage to disregard the disparity value. Disparities corresponding to high-confidence values could be passed onto a two dimensional resistive grid, as will be discussed in the following section.

(b) Monocular and occlusion points: As described in Section 1.2.3, these are points in the image visible to only one eye. Not all monocular points are due to occlusion. Occlusion is a complicated cue especially for signaling depth discontinuities and grouping surfaces. The Mahowald approach goes far by introducing monocular cells, whose behavior mimics that of tuned inhibitory cells, which break the interpolating resistive fuse between matching elements, causing the surfaces that intersect at the occlusion point to break away from each other. My implementation allocates this problem to the confidence measure and includes no specific processing for occlusion. Hardware test shows occlusion points coincide with low-confidence points.

(c) The aperture problem related to windowing: No image processing algorithm can process data it can not access or "see." Figure 4.1 illustrates an example. If the visual field were restricted to pixels 7-67, all visual algorithms, local as well as global, would fail. That is because within that range, the left image contains two more targets than the right image. Pixels 1 and 73 provide the necessary information for resolving this ambiguity. With local algorithms using windows there are additional restrictions. Suppose that a window five pixels wide was centered around pixel 37 in the left image and the search algorithm was combing the right image between pixels 29 and 45. There are three perfect matches within this region. In the absence of a clear maximum, my implementation would signal low-confidence.

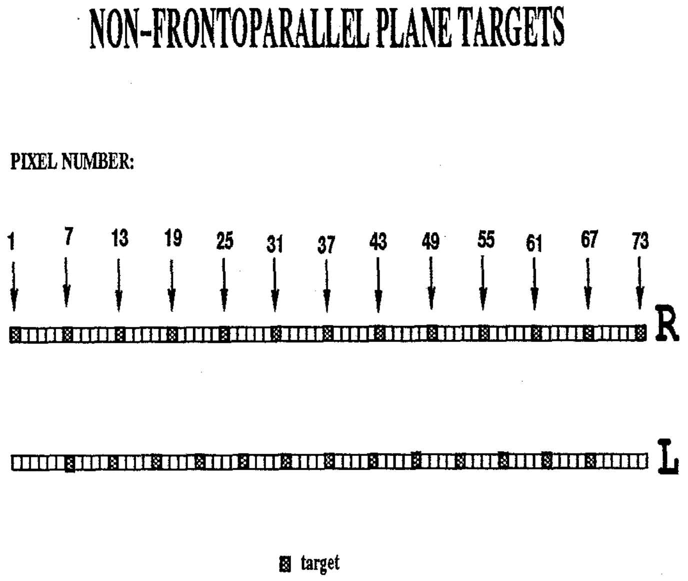


Figure 4.1: The aperture ambiguity

4.2 Description of current implementation

The architectural overview of the overall proposed system is shown in Figure 4.2. All elements shown have been implemented in VLSI except the area enclosed inside the ellipse where the disparity is smoothed using a resistive grid. Disparity is computed only along each horizontal scan line. An extension of the architecture described here, requiring more VLSI area and I/O pins, could be used to solve a restricted two dimensional problem where the epipolar line can be corrected for as well. Simulations with such an architecture were discussed in Chapter 3, Section 3.2.2.

The sections below contain the details of the implementation in VLSI:

4.2.1 Initial filtering

Prior to processing, the images are filtered by a one dimensional resistive grid. This smoothes the image to the appropriate spatial scale and, to some extent, reduces the undesirable effects of noise. In simulations of the previous chapter, the resistive grid was

1-D STEREO CIRCUIT ARCHITECTURAL OVERVIEW

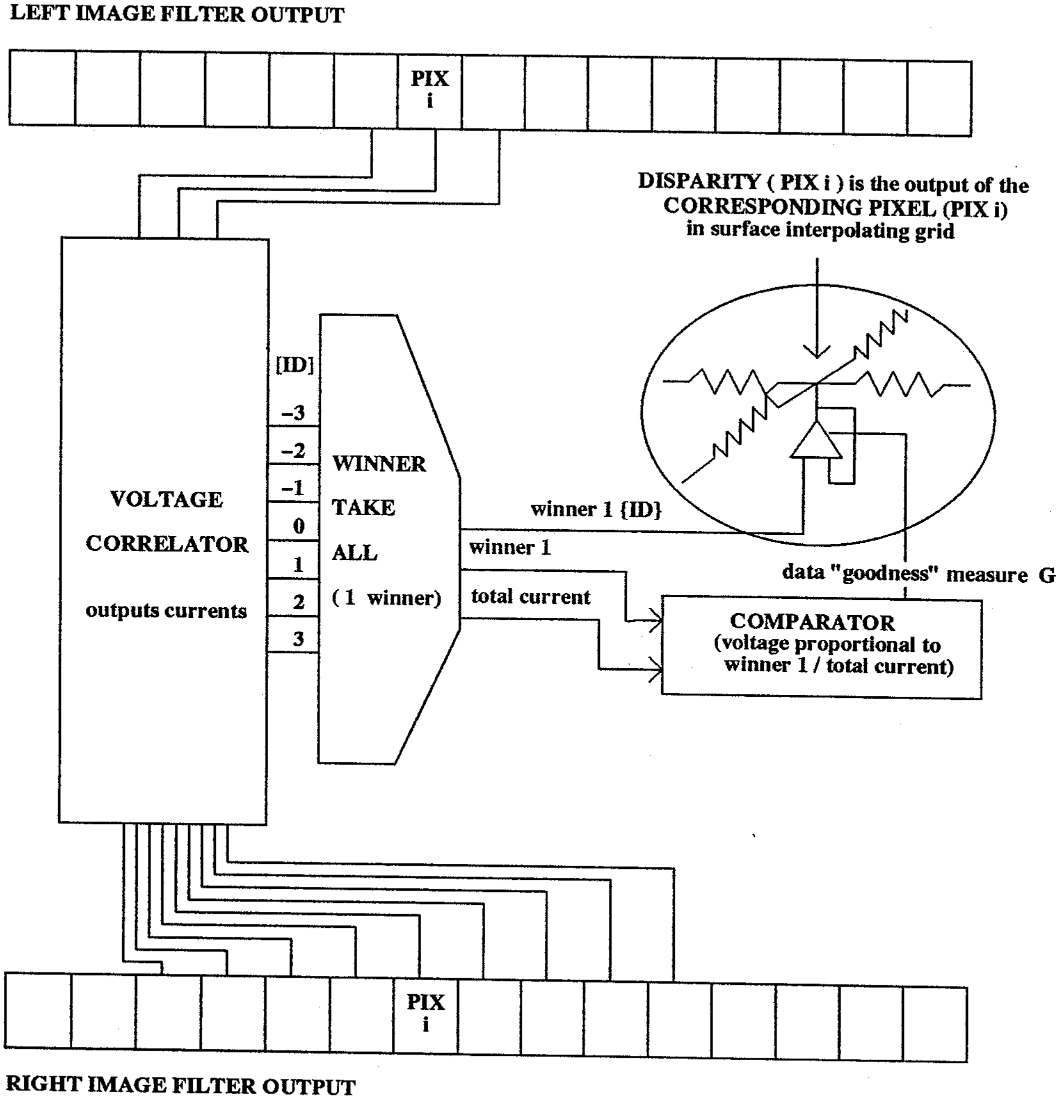

LEFT IMAGE FILTER OUTPUT
PIX i
DISPARITY (PIX i) is the output of the CORRESPONDING PIXEL (PIX i) in surface interpolating grid
VOLTAGE CORRELATOR outputs currents
[ID]
-3
-2
-1
0
1
2
3
WINNER TAKE ALL (1 winner)
winner 1 {ID}
winner 1
total current
data "goodness" measure G
COMPARATOR (voltage proportional to winner 1 / total current)
PIX i
RIGHT IMAGE FILTER OUTPUT

Figure 4.2: Analog VLSI Architecture for Stereopsis

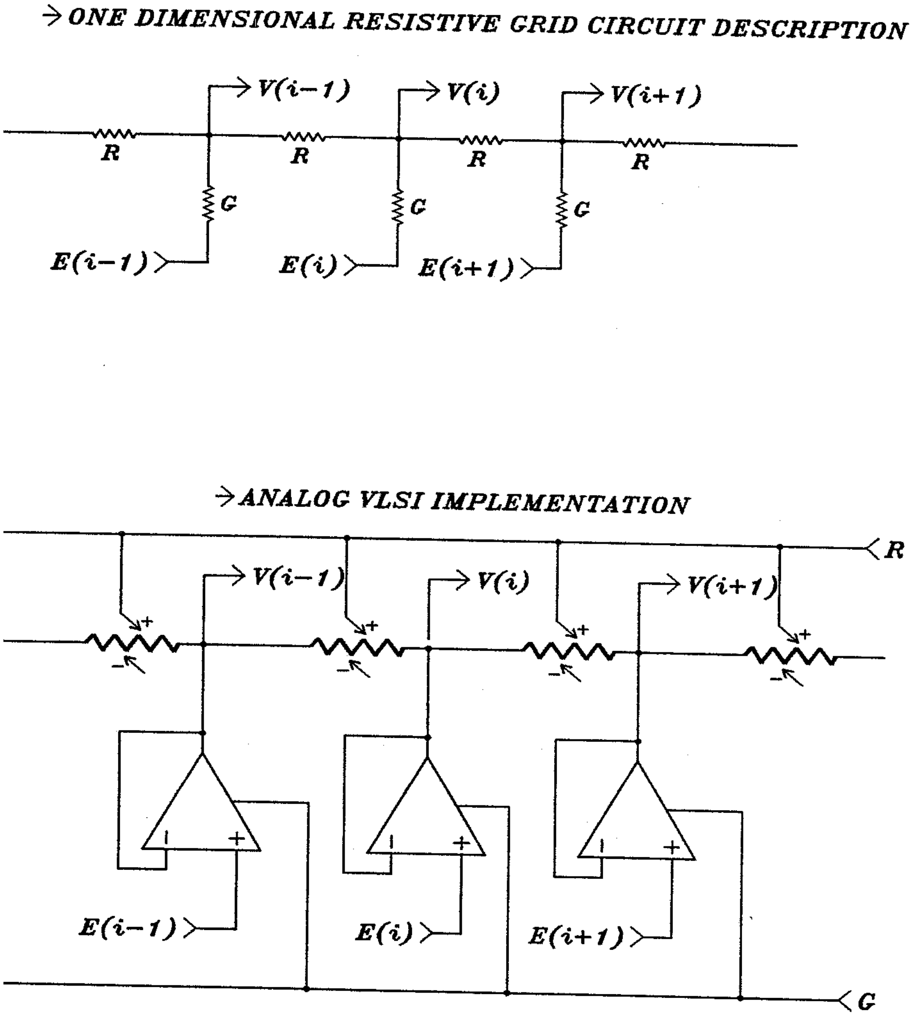
approximated by an exponential filter. Thus, the length of the resistive grid in hardware is analogous to the σ value of the exponential filter in software simulation.

The one dimensional resistive grid (Figure 4.3) has been implemented in two ways in hardware. Implementation (I) uses the horizontal resistor (HRes) circuit [Mead89b] to form the horizontal component of the resistive grid. This is a circuit that for a certain range of ΔV across its terminals acts like a linear resistor, and once the range is exceeded, turns into a current source. The range of linear operation is quite narrow: $\Delta V < 0.5$ V. The resistance and the range are controllable within certain limits. Typical values of the range in subthreshold operation are ±100 mV. The resistance in the linear range is around 10^9 to $10^6 \Omega$. The vertical resistors are formed by connecting the transconductance amplifier [Mead89b] in the follower configuration. The conductance (G) can be controlled by setting the current of the bias transistor. Again within a certain range of ΔV, this device acts like a linear resistor. Outside that, it is essentially a current source.

Implementation (II), on the other hand, uses gate polysilicon, snaked over the substrate to form the horizontal resistor. The value selected was $10^4 \Omega$. The value is below the minimum value obtainable with the HRes circuit. Thus, experimentation over a wider range of horizontal resistance values is possible. Besides, given the resistivity of polysilicon (around 20 Ω/square), any larger value would take too much area.

Difficulties with this circuit are caused mainly by mismatches. In implementation (I), the controls for range and resistivity of the HRes circuits are global. Since one HRes circuit is not identical to another in transistor size, parasitic capacitance, etc., there will be variations among the resistances, saturating currents, and to a lesser extent, linear ranges of HRes circuits. In implementation (II), the manufacturing process will not always yield a uniform width of polysilicon, causing variations in the values of the resistances. In both implementations, the values for G are not expected to be uniform across the chip, either.

There are 19 input pins onto the one dimensional resistive grid dedicated to the right image (to terminals labeled E_i in Figure 4.3). The actual grid is long enough to accommodate 23 inputs. The three end terminals are tied together to terminate the edges of


→ONE DIMENSIONAL RESISTIVE GRID CIRCUIT DESCRIPTION

→ANALOG VLSI IMPLEMENTATION

Figure 4.3: One dimensional resistive grid on chip

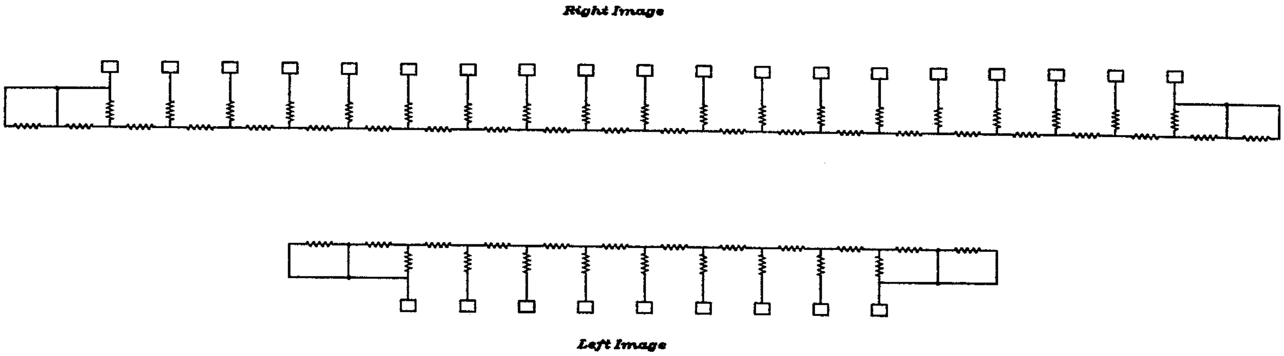


Figure 4.4: Input pin layout for the two one dimensional grids

the grid gracefully. The same strategy has been applied to the resistive grid for the left image. There are 9 inputs onto a grid capable of handling 13 (Figure 4.4).

4.2.2 Implementation of the hardware similarity metric

As in simulation, the chip uses "filtered" pixel values from the resistive grid to compare a window of the left image with the same size window from the right image. This comparison is made at each possible disparity value. The similarity value for a given disparity is the summed current of the 'bump' circuits [Delbrück91] corresponding to that disparity. Assuming that the window is $(2\gamma + 1)$ pixels wide, this current can be written as the following sum:

$$I_{out}(x,\delta) = \sum_{i=x-\gamma}^{i=x+\gamma} \frac{I_{bias}}{1 + \frac{4}{w}cosh^2(\frac{\kappa}{2kT} * (Image_R(i) - Image_L(i-\delta)))} \tag{4.1}$$

Where I_{bias} is the current in the bias transistor of the bump circuit, w and κ are circuit parameters, kT is a constant, δ is the disparity, and $Image_R(x)$ and $Image_L(x)$ are filtered pixel values of the right and the left image respectively.

In the VLSI implementation, the window width is five pixels (i.e.,$\gamma = 2$).

There are 11 current sums of the kind shown in Equation 4.1. These correspond to disparities in the range $[-5, 5]$. Figure 4.5 shows the circuit implementation.

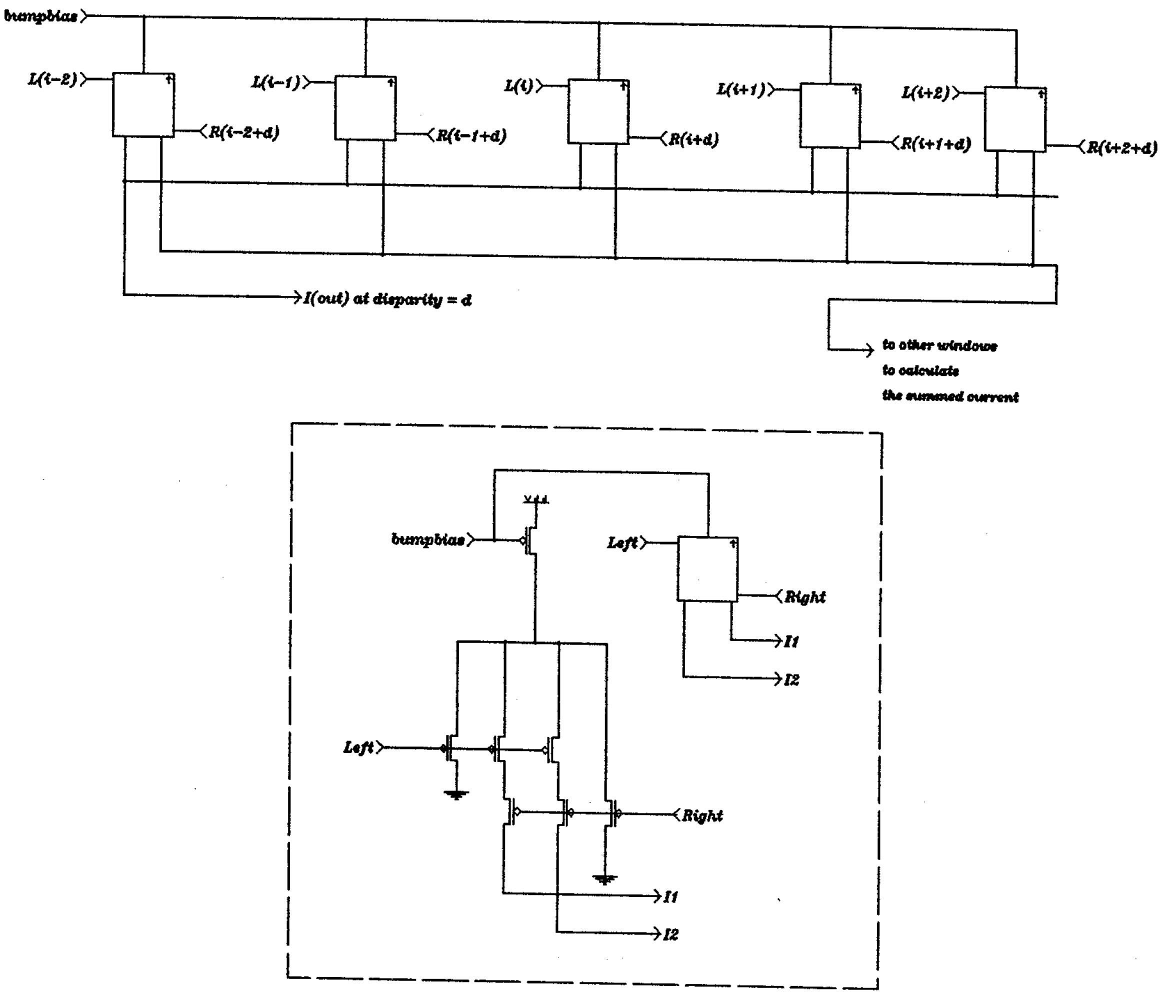


Figure 4.5: Bump window corresponding to $disparity = d$

Predictably, the practical implementation of this circuit will be plagued by mismatches. Slight variations in the circuit parameter w and in the sizing of other transistors, particularly the bias transistor, will reflect in I_{out}. This will be especially important in trying to determine the maximum current correctly between competing windows, as we shall see in the next section.

4.2.3 Determining the disparity

Eleven currents, one from each window, are input to a winner-take-all (W-T-A) circuit [LRMM89]. The current corresponding to the window that generates the highest current sum is the winner and determines the disparity value at the current pixel. Assuming that the allowed disparity range is between $-\Delta$ and Δ, this can be written as:

$$Disparity(x) = \delta \quad : \quad I_{out}(x, \delta) = \max_{-\Delta \leq \xi \leq \Delta} I_{out}(x, \xi) \quad \leq \frac{w(2\gamma + 1)}{w + 4} I_{bias} \qquad (4.2)$$

The last inequality is included to show that the current (and consequently the value of the metric) is limited to a fraction of I_{bias}. This property could be exploited in computing the confidence metric, as well as in determining monocular regions.

In the VLSI implementation $\Delta = 5$. In simulations, this proved to be sufficient for 64 x 64 images. Hardware test results will follow.

The maximum current typically generates a voltage between 2.0 – 2.8 V. The rest of the currents generate voltages close to 0 V. Thus, these voltages can be used to set the conductances of a series of followers that are connected to a tilted voltage line, as shown in Figure 4.6. The follower connected to the winning current will set the voltage on the common output node. This voltage will carry the disparity information. In the chip, the ends of the tilted voltage line were connected to the power rails. In this voltage divider configuration, each disparity has its own assigned voltage. The range is 1.25 V for the maximum negative disparity (-5) to 3.75 V for the maximum positive disparity (+5).

There is an added benefit that emerges from this configuration. Generally, there is only one winner because of the characteristics of the winner-take-all circuit. In rare instances

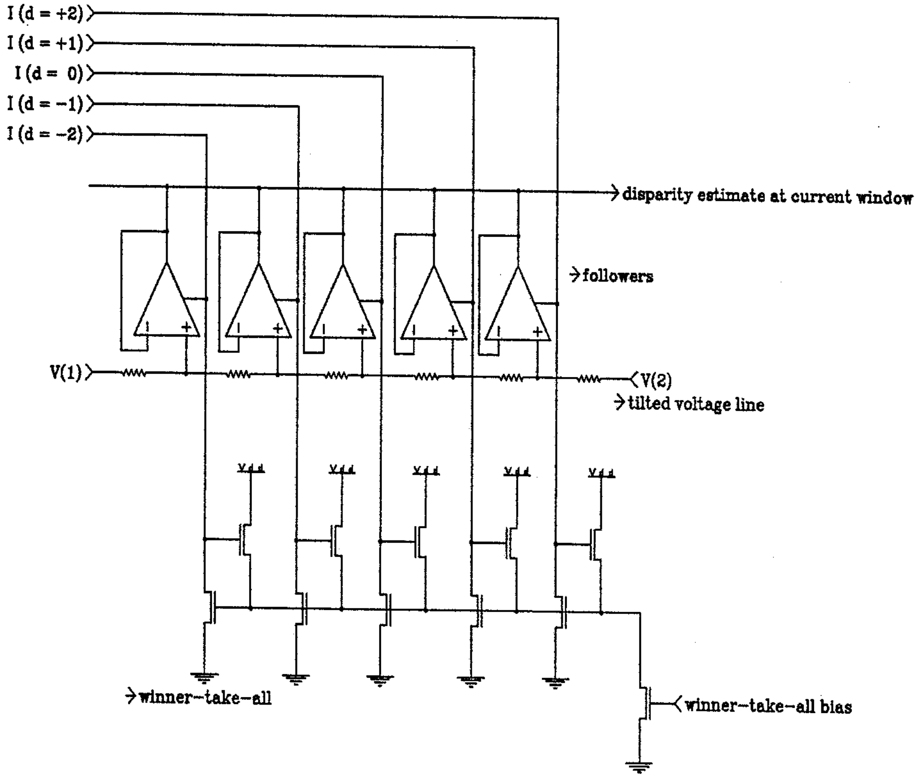


Figure 4.6: Winner-take-all circuit and disparity estimation

that currents are too close in value, the disparity will be the weighted average of the inputs from all winning followers. The exact value of V_{out} is:

$$V_{out} = \frac{\sum_i G_i V_i}{\sum_i G_i} \tag{4.3}$$

Since multiple winners are expected to be clustered around one disparity value, the output will be their average and most often the correct disparity.

The winner-take-all circuit is quite robust. The main difficulty with this circuit comes from the previous stage of bump circuits (Section 4.2.2). Due to circuit mismatches, the window that generates the maximum current need not be the window that is the correct match. Circuit simulations show that this is a likely outcome, the probability of which decreases with larger window size: Because more currents are summed, the impact of variations get averaged out. In hardware testing, this particular problem was observed when the image area to be matched did not have much variation or when the image was over-smoothed (Section 4.3.3), making the determination of a single maximum current difficult. The circuit then seemed to prefer one disparity value, most likely set by the window with the maximum hardware default current.

4.2.4 Confidence values

In areas of the image with flat intensity values (i.e., no features), the comparison of bump circuit currents will not produce a clear maximum. To make matters worse, the physical implementation introduces further ambiguity (Section 4.2.3).

Under such ambiguous conditions, the maximum current (and consequently the disparity) will be arbitrary.

Most computational approaches, in the absence of sufficient feature information (or targets), introduce window size adjustment to include enough targets for a meaningful match. To avoid this adjustment, which is difficult in hardware, my algorithm introduced a "confidence" measure. When this is below an adjustable threshold, low-confidence is reported.

Low confidence is also reported at occlusion points or when the window size is inadequate for resolving ambiguities.

The confidence measure is determined by a very simple arithmetic computation, namely the ratio between the maximum current and the sum of all currents (Chapter 3, Section 3.1.2).

$$Confidence(x) = \frac{\max_\delta I_{out}}{\sum_\delta I_{out}} \tag{4.4}$$

This computation contains a division, which is not easy to implement in analog VLSI. Instead, a current fractioning method was designed. Since the value of the ratio we are trying to compute is always less than 1, thresholding a fraction of $\sum_\delta I_{out}$ with the maximum current $\max_\delta I_{out}$ serves a similar function: Instead of trying to divide a current by another, we take an adjustable fraction of the larger current and compare it to the smaller one. I will start from the basics in describing the details. In the subthreshold region the current through a transistor is given by:

$$I = I_0 e^{\kappa V_g}(e^{-V_s} - e^{-V_d}) \tag{4.5}$$

where all voltages are scaled by kT/q and are with reference to the bulk.

For sufficiently large V_{ds}, $e^{-V_d} << e^{-V_s}$ and the current I can be scaled by changing V_{gs}. This can be done by implementing a current mirror, where we scale the mirrored current (i.e., V_g is the same between transistors), by increasing V_s with reference to the bulk (Figure 4.7). We then compare the fractioned current sum to the maximum current and adjust V_s until the desired operating point is reached. Thus, not only is the confidence circuit an extension of the winner-take-all architecture, but also the confidence value is computed in parallel with the disparity estimate. The resulting circuit is a compact integrated structure (Figure 4.8).

The pin for confidence value is near 0 V when the disparity output carries a high confidence and near 2.0 V when it carries a low confidence. In smooth areas of the image as well as at distinct occlusion points, low confidence is reported.

Thus, disparity output can be regarded as a sparse map with gaps corresponding to the low confidence points. A dense map can be obtained from the sparse values by interpolating between the high confidence values. This operation is very suitable for a surface

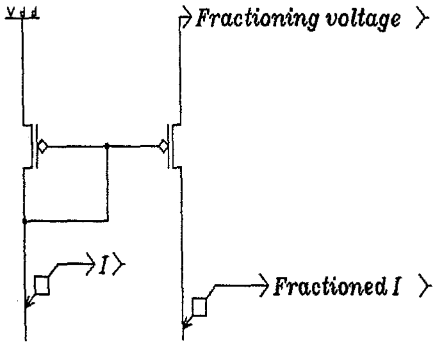


Figure 4.7: Fractioning current mirror

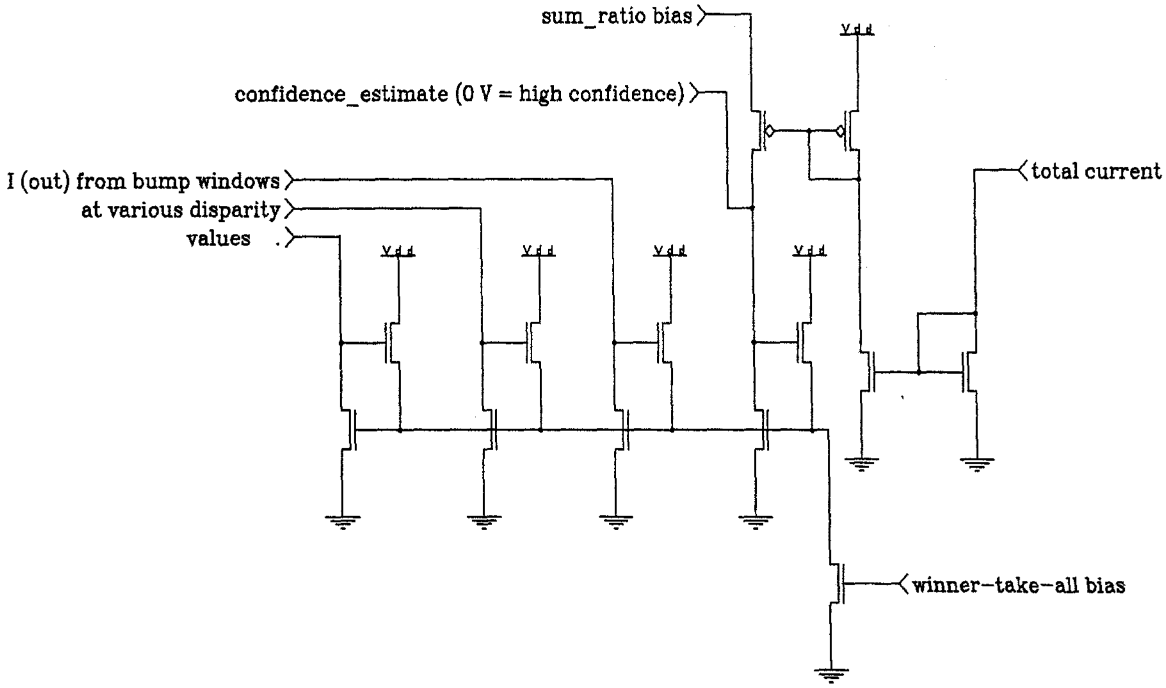


Figure 4.8: Confidence circuit as part of the winner-take-all structure

interpolating resistive grid, where the confidence determines the conductance (G) through which the disparity is input onto the grid.

This hardware implementation stops at outputing the disparity estimate and the confidence value for the window under evaluation. It does not contain the surface interpolating grid.

4.3 Test results

4.3.1 Setup

The chip does not contain any scanners. Therefore, all inputs from images (19 from the right image and 9 from the left) are input in parallel. Each input in time corresponds to a single window centered around a single pixel. Let's suppose that we want to match two 64 x 64 images. We can input the entire right image, but we have to trim the two sides of the left image to be able to search across all possible disparities. It turns out that a 64 x 64 image corresponds to 2944 input instances, corresponding to 64 rows and 46 columns due to this trimming effect. The pixel around which the sliding window is centered moves left to right across a row. Once a row is processed, this pixel moves to the leftmost column of the next row.

The chip contains five adjustable parameters:

1. R value: This sets the value of the horizontal resistors in the one dimensional resistive grid. In implementation (II) this is fixed at $10^4\Omega$. In implementation (I) it is controllable by adjusting the gate voltage of bias transistor of the HRes circuits (Figure 4.3).

2. G value: This sets the value of the vertical resistors in the one dimensional resistive grid. Its adjustment varies the gate voltage of the bias transistor of the followers (Figure 4.3).

3. W-T-A bias: This value determines the gate voltage on the transistor that biases the W-T-A circuit, and consequently its current capacity (Figure 4.6).

4. Bump circuit bias: This value determines the gate voltage on the bump circuit bias, and consequently its current capacity, I_{bias}, in Equation 4.1 (Figure 4.5).

5. Confidence bias: This value determines what fraction of the summed current will be compared to the maximum bump circuit window current (signal sum ratio bias in Figure 4.8).

In addition, input data limits can also be adjusted. Reasonable values range between 0.75 V and 4.25 V. Most tests, however, were carried out using a smaller range 1.5 V to 3.5 V with the thought of accommodating the silicon photoreceptor [Mead89b].

For testing the chip was connected to a custom board with a PC interface. The board converts digital input from the PC to analog input to the chip. Similarly, it also converts the analog chip output to digital representation for PC storage.

Input files were stored in the hard disk of the computer. A *C* program was used to input each line of the file to the chip and store the disparity and confidence values from the chip in a file. These values were converted to image files to form the disparity and confidence maps in the following sections.

4.3.2 One dimensional signals

Two tests with one dimensional images were prepared to demonstrate the chip's ability to resolve ambiguities. All images in this category are binary. Voltages corresponding to black and white pixels for the test are 3.5 V and 1.5 V respectively.

Occlusion image

Image contains two regions, one with zero and the other with constant negative disparity separated by an occlusion point. There is an additional occlusion point to the left of the leftmost target. Figure 4.9 shows the targets and the results from implementation (I). The chip determines the correct disparity; in addition, it signals low confidence at the following locations:

1. Where no targets are present, at right and left edges of the image.

2. At the two occlusion points (circled in Figure 4.9).

3. At seemingly regular points, where the input has a particular pattern. This is most likely due to undesirable circuit behavior. Due to transistor mismatches a certain combination of inputs fails to produce a clear winning current. Adjustments of circuit parameters, especially the confidence bias, could alleviate such a problem.

The plotted disparity response was obtained by first discarding all low confidence disparity values and then interpolating between the high confidence disparity values.

Tilted surface

Image contains targets from a non-fronto-parallel surface. Disparities span the whole range of the chip, from -5 to 5. The surface is marked with targets at quasi-regular intervals. Figure 4.10 shows the targets and the chip response to this input. Low confidence is reported again when no targets are present. As with the previous image, possibly due to circuit mismatches, the chip indicates no confidence points at regular intervals when given a particular pattern.

As before, the resulting plot was obtained by interpolating between the high confidence disparity values.

4.3.3 Random dot stereograms (RDS's)

Various experiments were conducted to evaluate the performance of the hardware with random dot stereograms. The test results confirm expectations and compare favorably with simulation results of the previous chapter.

Results with implementation (I)

G and R values determine the diffusion length of the one dimensional resistive grid, as shown by the following formula for the spread of voltage V_0 in a one dimensional resistive grid (similar to that in Figure 4.3) [Mead89b]:

$$V = V_0 \, e^{-\frac{1}{L}|x|} \tag{4.6}$$

where

$$L = \frac{1}{\sqrt{RG}} \tag{4.7}$$

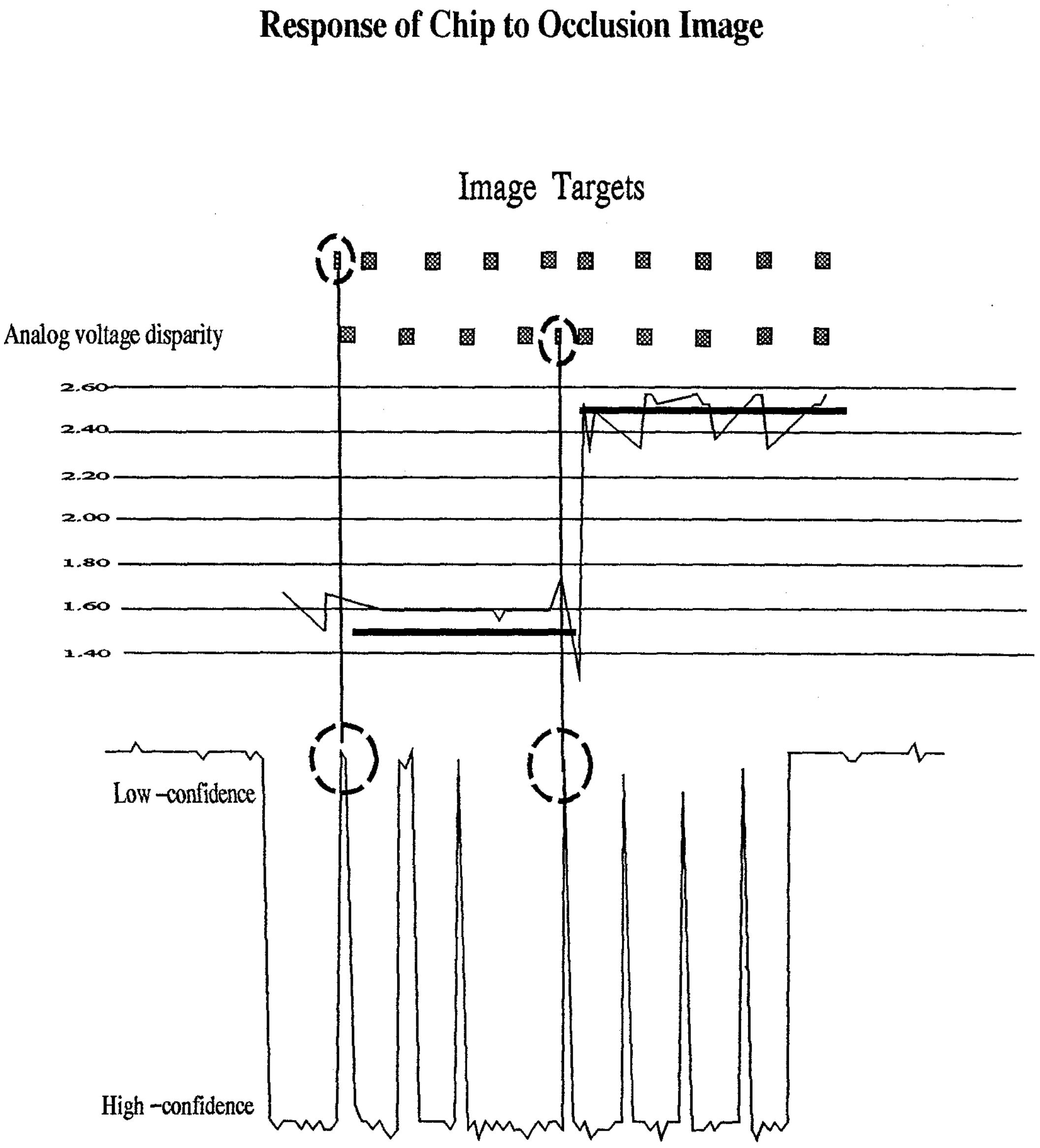


Figure 4.9: Occlusion image results with occlusion points circled

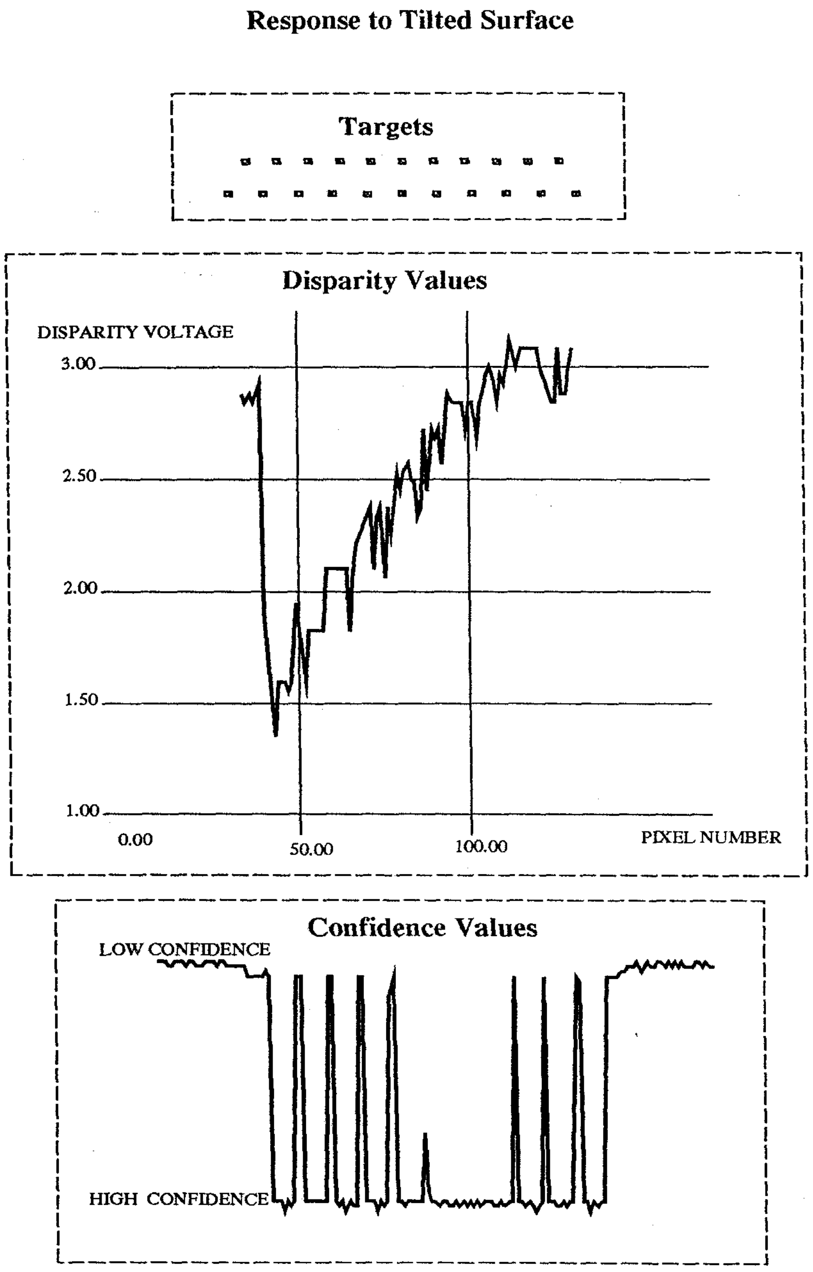


Figure 4.10: Response to tilted surface

In the discrete case where current is injected through each G onto the grid (exactly as in Figure 4.3), the equation for voltage at the node k becomes [Mead89b]:

$$V_k = \frac{1}{2G_0} \sum_n \gamma^{|n-k|} I_n \tag{4.8}$$

where

$$\gamma = 1 + \frac{1}{2L^2} - \frac{1}{L}\sqrt{1 + \frac{1}{4L^2}} \tag{4.9}$$

and

$$G_0 = \sqrt{\frac{G}{R}} \tag{4.10}$$

and again referring to Figure 4.3

$$I_n = \frac{E_n - V_n}{G} \tag{4.11}$$

In implementation (I), both the G and R values are adjustable. Tests were carried out holding G value constant (1.2 V) and varying the R value. Results are shown in Figure 4.11. R values from left to right are 0.5 V, 0.75 V, 1.0 V. No significant change is noted because the RG value is quite high in all settings.

I compared the chip output to the correct disparities. Figure 4.12 shows the error image with the perfect disparity image on the left. (If the chip output did not contain any error, the error image would be entirely gray.) Average error is around 0.25 V for all three chip outputs. The variance of the error σ_e^2 is also around 0.25 V.

Varying the G value in implementation (II)

In implementation (II), the R value is fixed around $10^4\Omega$. Thus decreasing the G voltage is equivalent to increasing the smoothing in simulation. Results are shown in Figure 4.13. From left to right $G = 4.0$ V, 3.5 V, 3.0 V, 2.5 V, 2.0 V and 1.5 V. It is clear that as diffusion length increases (i.e., smoothing increases), performance declines, especially below $G = 2.5$ V. At the rightmost setting the disparity map contains only noise. Error analysis results are show in Table 4.1.

Similar results were observed with simulation. As smoothing makes the stereo correspondence problem more ambiguous, more and more incorrect disparities and low confidence points are reported (Chapter 3, Section 3.2.2).

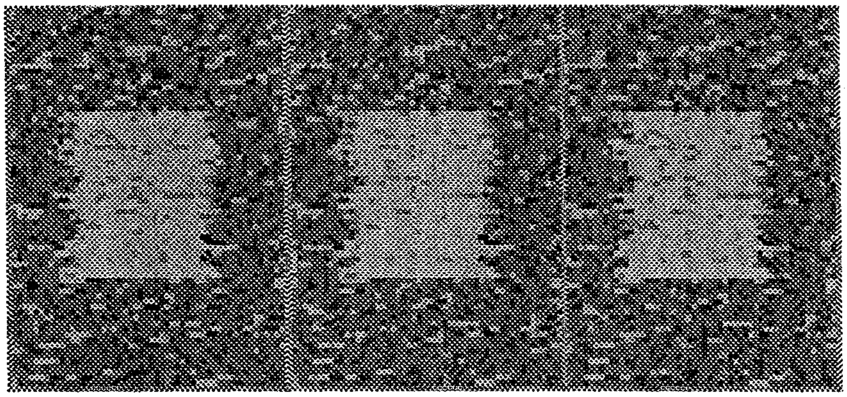


Figure 4.11: RDS results with implementation (I)

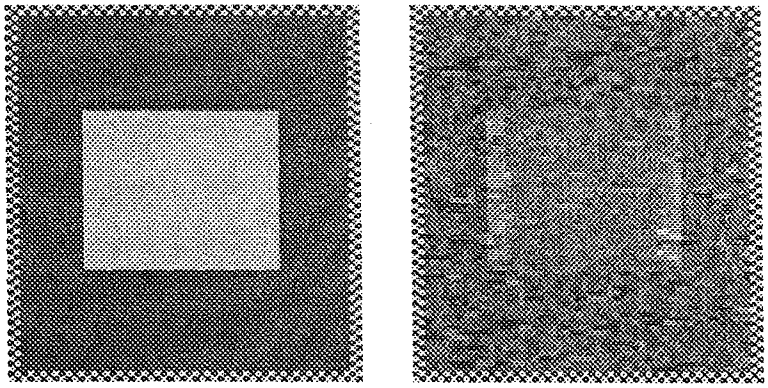


Figure 4.12: Error in RDS results of implementation (I)

$G\,(inV)$ $R = 10^4\Omega$	η_e	σ_e^2
4.0	0.12	0.43
3.5	0.12	0.45
3.0	0.13	0.47
2.5	0.14	0.53
2.0	0.15	0.64
1.5	0.14	0.80

Table 4.1: Error analysis with RG values

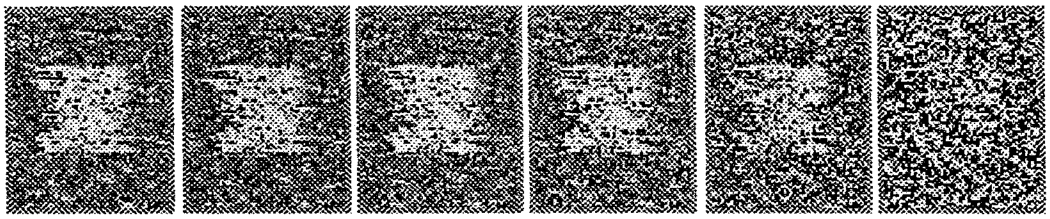


Figure 4.13: The disparity map fades as RG decreases

Same results can not be demonstrated with implementation (I) because the R value there is too high. To increase the diffusion length to a high enough value while keeping the G within reasonable limits is not possible.

Subthreshold settings

To demonstrate that chips from implementation (II) are still fully functional in spite of difficulties due to their low R value, I include the results in Figure 4.14. All parameters (except for G) were set at subthreshold values. Results on the left and right are from simulation and hardware test respectively.

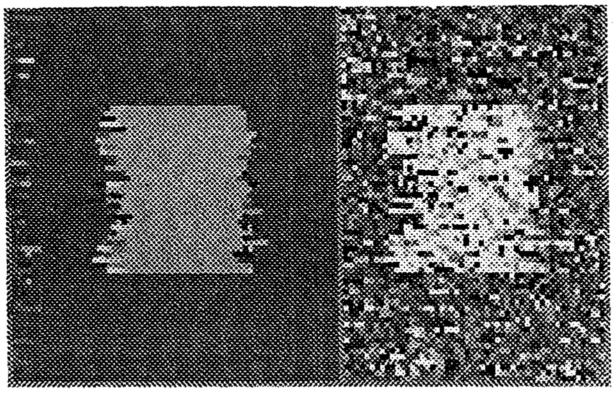


Figure 4.14: RDS results with implementation (II) subthreshold

Target Density (%)	η_e	σ_e^2
50	-0.25	0.25
30	-0.32	0.36
20	-0.34	0.38
10	-0.38	0.35
5	-0.45	0.38

Table 4.2: Error analysis with various target densities

Adjusting the target density with RDS's

In Chapter 3, Section 3.2.1, I demonstrated that decreasing the target density in an RDS causes the matching problem to become more ambiguous. I also did hardware experiments to show the performance of the chip with target densities from 50% to 5% (Figure 4.15). From left to right the target density values are 50%, 30%, 20%, 10% and 5%. It is clear that as was the case with the simulation results, as target density decreases, chip performance declines. Table 4.2 shows the results of error analysis.

4.3.4 The synthesized image pair

Information regarding this image pair was given in the previous chapter (Section 3.2.2). Image dimensions were reduced to 64 x 64 pixels to make it suitable for the hardware test.

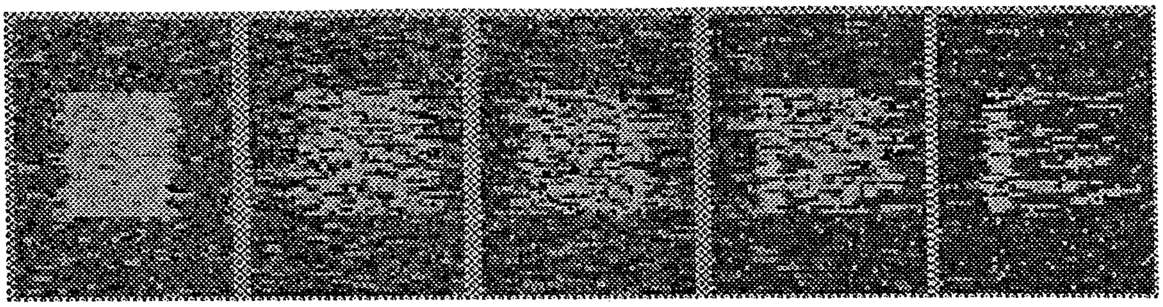


Figure 4.15: Adjusting the target density of an RDS

This reduction is necessary because disparity values with the original 128 x 128 image are outside the range of hardware. Resulting disparity maps are of dimensions 64 x 46 because of the trimming effect.

Tests were carried out with both implementation (I) and (II).

In tests with implementation (I), the results from which are better, the settings were above threshold. The 256 different gray levels of this image create a far more ambiguous matching problem than the binary values of an RDS. If, in addition, the current levels are set low (by setting the parameters subthreshold), performance degrades. Figure 4.16 shows the resulting disparity maps. The leftmost map is included for reference. It is the simulation result for the scaled image. Both the reduced resolution and the reduction process itself add to produce worse than usual results. The two disparity maps on the right were obtained from two different chips of implementation (I). Parameters were left unchanged between chips.

Error analysis was carried out for the hardware results. The correct disparity was approximated by the results from the two dimensional simulation in Chapter 3, Section 3.2.2. Figure 4.17 shows the correct disparity and the two error images obtained by taking the difference between the correct disparity image shown on the left and the two chip outputs in Figure 4.16. The statistical analysis of the error indicates that average error is between

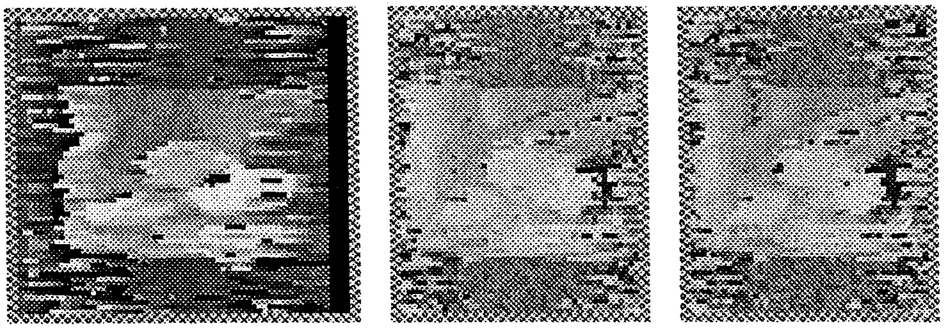


Figure 4.16: Synthesized image pair results with implementation (I)

0.10 - 0.25 V. The variance of the error, σ_e^2 is between 0.2 - 0.4 V.

In tests with implementation (II), the settings (other than that for G) correspond to subthreshold voltages. Performance degradation caused by using subthreshold settings is not significant compared to performance degradation caused by the small R value that over-smooths the input to make matching more ambiguous. Figure 4.18 shows results with two chips of implementation (II) with all hardware parameters left unchanged between chips. The leftmost map is identical to the simulation reference in Figure 4.16.

Figure 4.19 shows the correct disparity and the two error images obtained by taking the difference between the correct disparity image shown on the left and the two chip outputs in Figure 4.18. The statistical analysis of the error indicates that average error is less than that in implementation (I), around -0.1 V. The variance of the error (σ_e^2), however, is greater than that in implementation (I), around 0.5 V.

4.3.5 The rocks image pair

The information about this image was given in the previous chapter (Section 3.2.3). Image dimensions were reduced to 60 x 64 pixels to accommodate hardware test. Resulting disparity map dimensions are 60 x 46 pixels.

Figures 4.20, 4.21 and 4.22 show chip outputs.

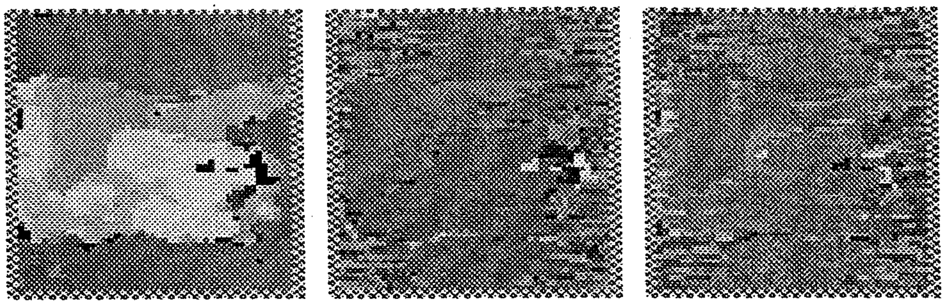


Figure 4.17: Synthesized image pair error images with implementation (I)

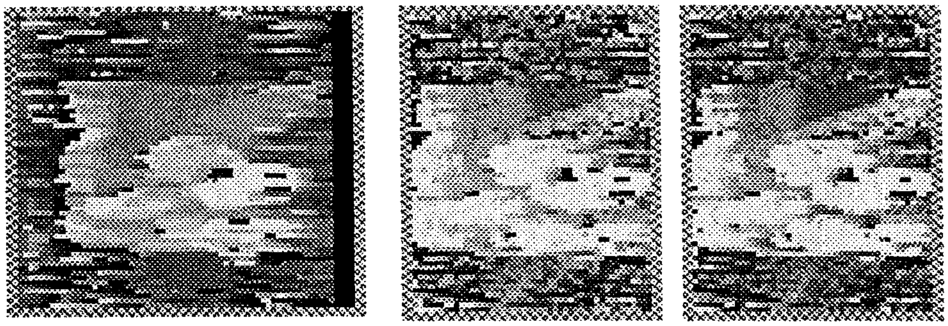


Figure 4.18: Synthesized image pair results with implementation (II)

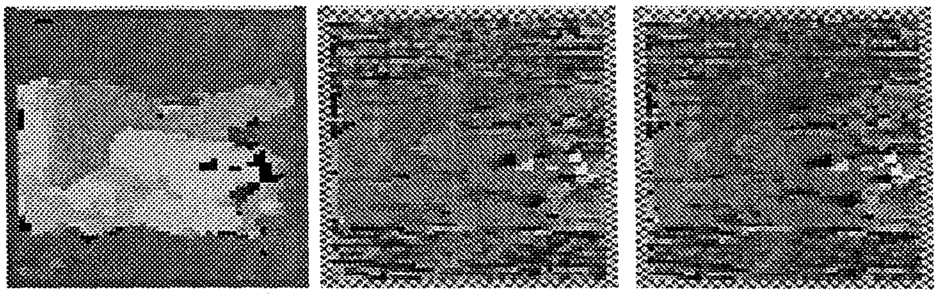


Figure 4.19: Synthesized image pair error images with implementation (II)

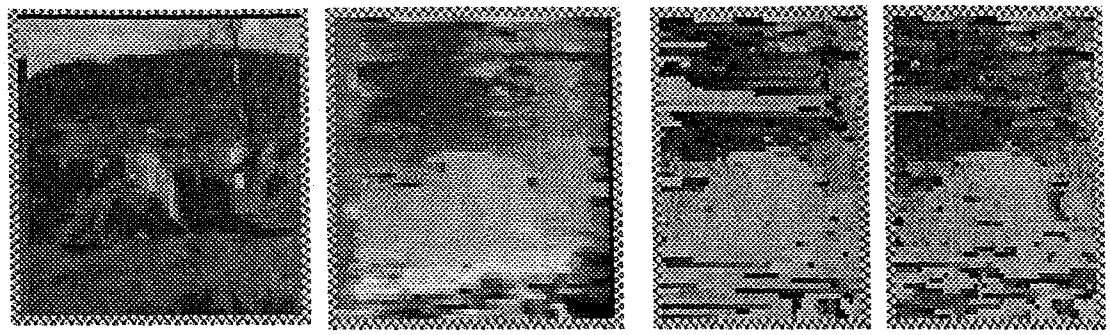


Figure 4.20: The rock image pair disparity output with implementation (I)

The two rightmost images in Figure 4.20 show the raw disparity outputs from two chips of implementation (I). The two leftmost images are the actual image and the disparity from simulation of the scaled rock image used during hardware test.

Figure 4.21 shows confidence values from the chips. Pixels shown in white are the high-confidence pixels. Note that areas with flat intensity are marked in black (i.e., low confidence). The original image is included for comparison.

Figure 4.22 shows processed disparity values in comparison with simulation results (leftmost image). Disparity is "interpolated" using only high confidence disparity values already computed.

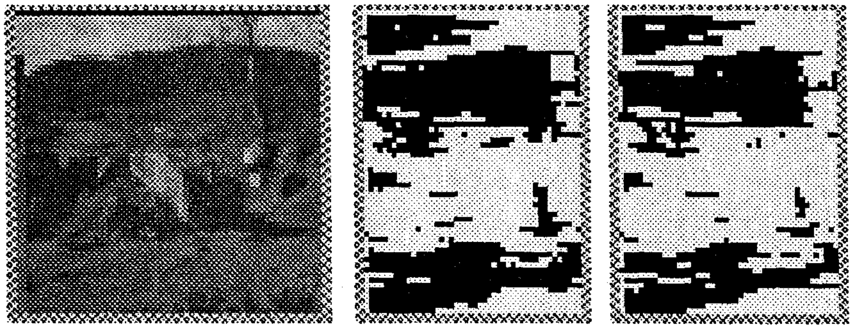


Figure 4.21: The rock image pair confidence output with implementation (I)

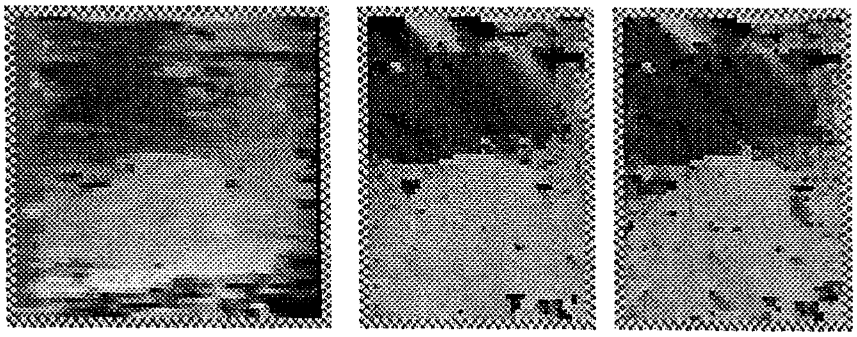


Figure 4.22: The rock image pair processed disparities

Chapter 5 Conclusions and Future Work

5.1 Summary

Work presented describes a hardware stereo correspondence algorithm, its hardware implementation and results obtained from simulation and hardware test. All of these collectively show that the system is functional, expandable to solve real-world problems in real-time, and implementable with existing technology.

5.2 Assessment of the Hardware Algorithm

5.2.1 Features

The algorithm possesses the following favorable features:

1. Simplicity: Considering the complexity of the stereo correspondence problem, the algorithm and its implementation are very simple. There is only one resolution scale involved. No post processing is needed and the disparity estimate and confidence at each pixel are computed in parallel. The serial nature of the algorithm makes it versatile: Large two dimensional images can be processed provided that the image disparity range is accomodated by the hardware disparity range.

2. Accuracy: Considering the simplicity of the algorithm, the disparity results are quite accurate. The confidence metric goes a long way towards compensating for the inherent ambiguities of the stereo correspondence problem.

3. Compact and low cost system: Many systems that perform similar tasks for solving equivalent vision problems require far more hardware at much higher cost. My stereo correspondence architecture could be implemented using two or three dedicated chips.

4. Low power consumption: This feature accompanies the previous one. The analog VLSI chips operating below or slightly above threshold consume far less power than large digital systems that emulate many vision algorithms.

5.2.2 Simulation

As I mentioned above, considering the simplicity of the algorithm, the disparity results are surprisingly good. The confidence metric improves the performance significantly at a rather small computational cost.

One criticism is that the confidence metric can not handle monocular and occluded points reliably even though it is quite effective for identifying flat intensity regions in images.

A second thresholding scheme involving only the maximum value of the hardware metric in the disparity window would be the most straightforward method for handling monocular and occlusion data.

5.2.3 Hardware

Hardware results were presented from two unique implementations of the algorithm. Among those, the first implementation proved to be more successful because *RG* values remained within the range necessary to keep the image from being over-smoothed prior to matching.

Hardware experiment results compared favorably with their simulation counterparts.

One shortfall with hardware experiments was that the synthesized and the natural images had to be scaled to accommodate the limitation in the disparity range. This scaling was shown to adversely affect resolution and the accuracy of simulation results. It is therefore highly likely that hardware performance could be improved if images that do not require scaling could be used. Random dot stereograms are such images. The results with random dot stereograms were indeed superior to the results with the synthesized image and the natural rock image.

5.3 Future Work

5.3.1 System applications

For practical applications of the stereo correspondence architecture presented, two distinct situations come to mind:

The one dimensional version of the chip (which has been implemented) is appropriate for non-convergent camera geometry. It can be used as part of the naviation system of an autonomous vehicle which utilizes two parallel video cameras (Figure 5.1).

The two dimensional extension of the chip is appropriate for use as part of the vision system of a robot manipulating objects within a close (several feet of the cameras) range. Such applications will generally require that the cameras be positioned to produce convergent geometry.

For applications that require higher accuracy, the chip can be made part of a larger network of circuits that use the disparity estimate provided by the chip as a rough estimate or starting point. Iterative schemes that draw from a series of disparity maps obtained from slightly perturbed the camera positions can be utilized to obtain a far more accurate disparity map of the scene.

Many applications will require the image-chip interface to be serial since parallel input of large, especially two dimensional neighborhoods will use too many I/O pins. This as along with the nature of camera output will necessitate the use of scanning circuits. Figure 5.2 shows schematically how this can be accomplished. Similar schemes have previously been designed and shown to be functional [MAG91].

5.3.2 Improvements to the algorithm and architecture

Multiple scales

The method of utilizing multiple resolutions was not explored with the hardware metric. As mentioned in Chapter 1, ambiguities in matching the two images can be readily resolved by using a coarse-to-fine matching strategy. This is very difficult to do in hardware:

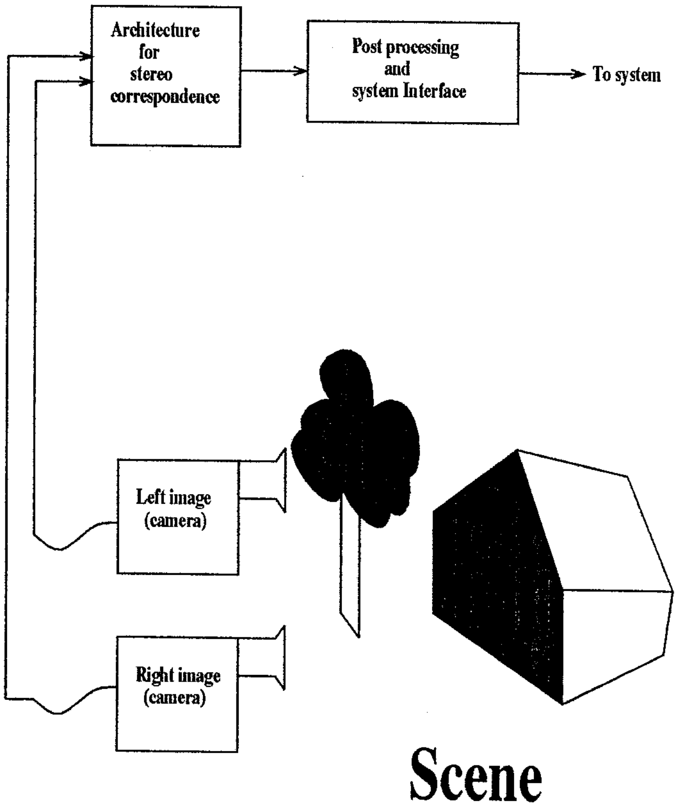

Architecture for stereo correspondence
Post processing and system Interface
To system
Left image (camera)
Right image (camera)
Scene

Figure 5.1: Application with parallel cameras

In an application using parallel cameras, assuming that vertical disparity is zero, the one dimensional version of the architecture can be used. Two camera outputs can be input to the chip using serial scanners (Figure 5.2). If convergent camera geometry is to be used, the two dimensional expansion of the architecture will be necessary.

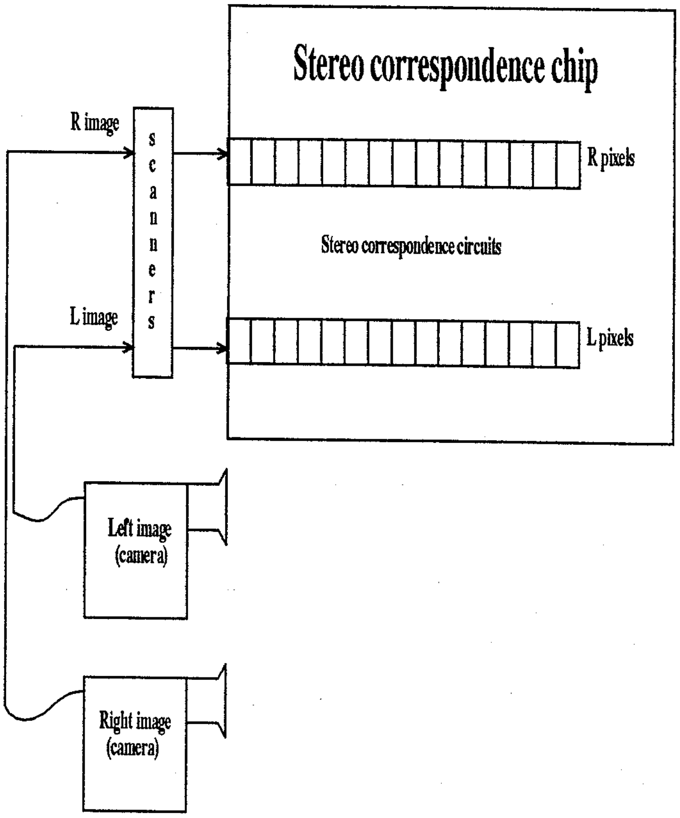


Figure 5.2: Scanners for serial I/O ports

Two camera outputs can be input to the chip using serial scanners. In the one dimensional case this is relatively simple. In the two dimensional case, however, further storage inside the chip or modification of the output protocol of the camera may be necessary.

Even the most straightforward hardware implementation would have to use a multiplexing scheme simultaneously with changing resolution and shrinking window of the image region and disparity range.

More reliable monocular and occlusion handling

In retrospect, improvement to handle monocular and occlusion points does not appear to require a lot of area on top of the existing hardware architecture. The most straightforward approach would be to compare the maximum current with an adjustable current to determine if the maximum current value is high enough to rule out the presence of occlusion and monocular points.

Including the second dimension

Simulation results obtained using a two dimensional matching region and a two dimensional match search area were presented in Chapter 3, Section 3.2.2. These showed a significant improvement over the one dimensional results. Improvement was most pronounced in the corners where images are likely to contain vertical disparity and along disparity discontinuities where jagged edges were a problem. Including the second dimension in hardware does not require any extensive design change to the existing architecture, merely an increased number of already described computation units, more VLSI area and I/O pins.

5.3.3 From stereo correspondence to motion correspondence

Once the second dimension is included, a major step is taken towards extending this stereo correspondence algorithm to a motion correspondence algorithm. There are, however, several restrictions that apply:

1. Only two frames can be processed at one time.

2. The image intensity gradient information ($\frac{\delta I}{\delta t}$) is not utilized.

3. Motion can be determined within a range. Movement outside these limits can cause the rest of the data to be unreliable, as well.

4. Two dimensional translational motion without any component perpendicular to the image plane will be relatively straightforward; translational motion involving the third dimension and rotational motion will require some post processing to assess correctly.

References

[Barnard86] (1986) S.T. Barnard. A stochastic approach to stereo vision. *Proceedings of the Fifth National Conference on Artificial Intelligence.* Philadelphia, 1986, 676-690.

[BB82] (1982) D.H. Ballard, C.M. Brown. *Computer Vision*, Prentice Hall, Englewood Cliffs, New Jersey.

[BBLR76] (1976) H.H. Bailey, F.W. Blackwell, C.L. Lowery, J.A. Ratkovic. *Image Correlation: Part I, Simulation and Analysis.* A report prepared for United States Air Force Project Rand, Santa Monica, California.

[BBP67] (1967) H.B. Barlow, C. Blakemore, J.D. Pettigrew. The neural mechanism of binocular depth discrimination. *Journal of Physiology*, **193**, 327-342.

[Berry48] (1948) R.N. Berry. Quantitative relations among vernier, real depth, and stereoscopic depth acuities. *Journal of Experimental Psychology*, **38**, 708-721.

[Chen88] (1988) C.C. Chen. *Markov Random Fields in Image Analysis.* Ph.D. Thesis in Computer Science. Michigan State University, East Lansing, Michigan.

[CG89] (1989) A.K. Chhabra, T.A. Grogan. Depth from stereo: variational theory and a hybrid analog-digital network. *SPIE Vol 1076 Image Understanding and the Man-Machine Interface II*, 131-138.

[CW77] (1977) P.G.H. Clarke, D. Whitteridge. A comparison of stereoscopic mechanisms cortical visual areas V1 and V2 of the cat. *Journal of Physiology*, **272**, 92-93P.

[Delbrück91] (1991) T. Delbrück. "Bump" circuits for computing similarity and dissimilarity of analog voltages. CNS Memo 10. California Institute of Technology Computation and Neural Science Memorandum.

[DR58] (1958) W.B. Davenport, W.L. Root. *An Introduction to the Theory of Random Signals and Noise* McGraw-Hill, New York.

[GCK91] (1991) A. Gruss, L.R. Carley, T. Kanade. Integrated sensor and range-finding analog signal processor. *IEEE Journal of Solid-State Circuits*, **26**, No. 3, 184-191.

[GG91] (1991) D. Geman, B. Gidas. Image analysis and computer vision. *Spatial Statistics and Digital Image Analysis* National Research Council, Panel on Spatial Statistics and Image Processing. National Academy Press, Washington, DC.

[GL76] (1976) W.L. Gulick, R.B. Lawson. *Human Stereopsis.* Oxford University Press, New York.

[Grimson81] (1981) W.E.L. Grimson. *From Images to Surfaces.* MIT Press, Cambridge, Massachusetts.

[Hannah74] (1974) M.J. Hannah. Computer matching of areas in stereo images. Doctoral Dissertation, Stanford University, Stanford, California.

[HKLM88] (1988) J. Hutchinson, C. Koch, J. Luo, C. Mead. Computing motion using analog and binary resistive networks. *IEEE Computer*, **21**, 53-63.

[HLLW91] (1991) J.M. Hakkarainen, J.J. Little, H.S. Lee, J.L. Wyatt. Interaction of algorithm and implementation for analog VLSI stereo vision. **SPIE 1473**, *Visual Information Procesing: From Neurons to Chips*, 173-184.

[Horn86] (1986) B.K.P. Horn. *Robot Vision.* MIT Press, Cambridge, Massachusetts.

[HS92] (1992) R.M. Haralick, L.G. Shapiro. *Computer and Robot Vision, Vol II.* Addison Wesley, Reading, Massachusetts.

[ITMMSH86] (1986) M. Iwashita, T. Temma, M. Mizoguchi, K. Matsumoto, M. Shuto, S. Hanaki. A data driven VLSI image processor (ImPP). *Evaluation of Microcomputers for Image Processing.* L. Uhr, K. Preston, S. Levialdi, M.J.B. Duff, editors. Academic Press, Orlando, Florida.

[JM75] (1975) B. Julesz, J. Miller. Independent spatial-frequency-tuned channels in binocular fusion and rivalry. *Perception*, 4, 125-143.

[JM92] (1992) D.G. Jones, J.Malik. A computational framework for determining stereo correspondance from a set of linear spatial filters. *European Vision Conference*, 1992.

[Julesz71] (1971) B. Julesz. *Foundations of Cyclopean Perception*, University of Chicago Press, Chicago, Illinois.

[Kosko92] (1992) B. Kosko. *Neural Networks for Signal Processing.* Prentice Hall, Englewood Cliffs, New Jersey.

[KSJ92] (1992) E.R. Kandell, J.H. Schwartz, T.M. Jessell. *Principles of Neural Science.* Third Edition. Elsevier Science Publishing Company, New York.

[Lange67] (1967) F.H. Lange. *Correlation Techniques*, London Iliffe Books LTD, Princeton, New Jersey.

[LRMM89] (1989) J.P. Lazzaro, S. Ryckebusch, M.A. Mahowald, C.A. Mead. *Winner-Take-All Networks of O(N) Complexity*, Caltech Computer Science Department Technical Report, Caltech-CS-TR-21-88.

[MAG91] (1991) A. Moore, J. Allman, R. Goodman. A real-time neural system for color constancy. *IEEE Transactions on Neural Systems*, **2**, 237-247.

[Mahowald92] VLSI Analogs of Neuronal Visual Processing. Doctoral Dissertation. California Institute of Technology, Pasadena, California.

[Marr82] (1982) D. Marr. *Vision.* W.H. Freeman and Company, New York.

[MD89] (1989) M. Mahowald, T. Delbrück. Cooperative Stereo Matching Using Static and Dynamic Image Features. *Analog VLSI Implementation of Neural Systems*, C. Mead, M. Ismail, editors. Kluwer Academic Publishers, Boston, Massachusetts.

[Mead89a] (1989) C. Mead. Adaptive retina. *Analog VLSI Implementation of Neural Systems*, C. Mead, M. Ismail, editors. Kluwer Academic Publishers, Boston, Massachusetts.

[Mead89b] (1989) C. Mead. *Analog VLSI and Neural Systems.* Addison-Wesley, Reading, Massachusetts.

[MKS89] (1989) L. Matthies, T. Kanade, R. Szeliski. Kalman-filter based algorithms for estimating depth from image sequences. *International Journal on Computer Vision*, **3**, 209-236.

[MP79] (1979) D. Marr, T. Poggio. A computational theory of human stereo vision. *Proc. of Royal Society of London*, B **204**, 187-217.

[OK85] (1985) Y. Ohta, T. Kanade. Stereo by intra- and inter-scanline search using dynamic programming, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **PAMI-7**, 139-154.

[OK89] (1990) M. Okutomi, T. Kanade. A locally adaptive window for signal matching. *Proceedings of the IEEE Third International Conference on Computer Vision*, 190-199.

[Papoulis65] (1965) A. Papoulis. *Probability, Random Variables, and Stochastic Processes* McGraw-Hill, New York.

[Parker85] (1985) I.N. Parker. VLSI Architecture. *VLSI Image Processing.* R.J. Offen, ed. McGraw-Hill, New York.

[PF77] G.F. Poggio, B. Fischer. Binocular interaction and depth sensitivity in striate and prestriate cortex of behaving rhesus monkey. *Journal of Neurophysiology*, **40**, 1392-1405.

[PMF85] (1985) S. Pollard, J. Mayhew, J. Frisby. PMF: a stereo correspondance algorithm using disparity gradient limit. *Perception* **14**, 449-470.

[PTK85] 1985) T. Poggio, V. Torre, C. Koch. Computational vision and regularization theory. *Nature*, **317**, 314-317.

[RDMDMS91] (1991) S. Ryckebusch, T. Delbrück, M. Mahowald, S. DeWeerth, M.A. Maher, M. Sivilotti. *Analog VLSI Laboratory Course.* Accompanies [Mead89a]. Pasadena, California.

[RFPST90] (1990) R. Regan, J.P. Frisby, G.F. Poggio, C.M. Schor, C.W. Tyler. Perception of stereodepth and stereomotion. *Visual Perception: The Neurophysiological Foundations.* L. Spillman, J.S. Werner, editors, 317-347.

[SMM87] (1987) M. Sivilotti, M. Mahowald, C. Mead. Real-time visual computations using analog CMOS processing arrays. *Proceedings of the Stanford Conference on Very Large Scale Integration*, MIT Press, Cambridge, Massachusetts.

[SS92] (1992) N.N. Schraudolph, T.J. Sejnowski. Competitive Anti-Hebbian Learning of Invariants. *Advances in Neural Information Processing Systems*, vol 4, 1017-1024.

[Szeliski89] (1989) R. Szeliski. *Bayesian Modeling of Uncertainty in Low-level Vision.* Kluwer Academic Publishers, Boston, Massachusetts.

[VK92] (1992) D. Van Essen, C. Koch. Vision (CNS/Bi 186) Class Notes 1992, California Institute of Technology, Pasadena, California.

[Weng90] (1990) J. Weng. A theory of image matching. *Proceedings of the IEEE Third International Conference on Computer Vision*, 200-209.

[Wesseley76] (1976) H.W. Wesseley. *Image Correlation: Part II, Theoretical Basis.* A report prepared for United States Air Force Project Rand, Santa Monica, California.

[WK86] (1986) J.A. Webb, T. Kanade. Vision on a Systolic Array Machine. *Evaluation of Microcomputers for Image Processing.* L. Uhr, K. Preston, S. Levialdi, M.J.B. Duff, editors. Academic Press, Orlando, Florida.

[WTK87] (1987) A. Witkin, D. Terzopoulos, M. Kass. Signal matching through scale space. *International Journal on Computer Vision*, **1**, 133-144.

Glossary

binocular involving both eyes.

cone cells high acuity, chromatic cells lining the eye's receptor sheet.

Cones mediate color vision and provide greater spatial and temporal resolution. They are concentrated in the fovea and saturate in their response only in intense light.

contour the monocular or binocular perception of a well-defined and continuous target boundary.

convolving an image I with a kernel k, having support or domain K, produces a convolved image, $(I * k)$, that is defined by

$$(I * k)(x, y) = \sum_{(i,j) \in K} I(x - i, y - j)k(i, j)$$

Convolution is a linear operator.

correlating an image I with a kernel k, having support or domain K, produces a correlated image J, defined by

$$J(x, y) = \sum_{(i,j) \in K} I(x + i, y + j)k(i, j)$$

degree of visual field area in the visual field corresponding to a single degree of an arc.

One degree of visual field is roughly the width of your thumb at arm's length.

depth map an image in which the value in each pixel's position is the distance between the image plane and the surface patch being imaged corresponding to that pixel.

disparity the difference in the positions of the images of the same three dimensional point in two perspective projection images taken from different positions.

epipolar line on one stereo image corresponding to a given point in another stereo image is the perspective projection on the first stereo image of the three dimensional ray that is the inverse perspective projection of the given point from the other stereo image.

fovea area of the retina normally corresponding to the center of the visual field, lined almost exclusively with high acuity cone cells and thus affording acute vision.

Gestalt psychology movement in psychology which started in the early twentieth century.

The central theme is that the act of perception creates a *Gestalt*, i.e., figure, form or image, beyond what is being perceived, which represents the *organization* of sensations in the brain.

gray level a number or value assigned to a position in an image.

This value is proportional to the integrated output response in a small area of the optic or photographic sensor that captures the image.

hyperacuity ability to carry out a variety of tasks to accuracies more precise than the dimensions of the retinal cones from which the information originates.

Foveal cones have a diameter of about 27", yet many tasks yield accuracies of around 5". Stereoscopic acuity may be as good as 2". Such tasks are said to fall within the range of hyperacuity.

image a spatial representation of an object, of a two dimensional or a three dimensional scene, or of another image.

image intensity gray level.

image matching the process of determining pixel by pixel, arc-by-arc, or region-by-region correspondence between two images taken of the same scene but with different sensors, different lighting, or a different viewing angle.

image smoothing any spatial filtering that spatially simplifies and approximates the input image, suppressing small details and enhancing large or coarse image structures.

kernel function defined on the domain of the linear spatial filter it represents, whose value at each pixel (of the domain) is the weight or coefficient of the linear combination that defines the spatial linear filter.

light striping a technique of projecting a light pattern on a scene, which is composed of successive planes of light that are all parallel.

The scene is then viewed from different directions. The pixels that image a surface patch lit by a known light pattern contain enough information to determine the three dimensional coordinates of the image patch.

linear spatial filter an image operator for which the image intensity at coordinates (x, y) is a weighted average or linear combination of the image intensities located at a particular spatial pattern around coordinates (x, y) of the input image.

monocular involving only one eye.

Müller's horopter zero disparity surface.

Horopter changes when the eyes or camera positions change.

occluding edge (boundary) an image edge (boundary) arising from a range or depth discontinuity in the scene.

occlusion situation that arises from a surface being visible to only one eye, because of another surface occluding the light reflected from it from reaching the other eye.

parallax the observed positional difference of a projected three dimensional point on a pair of two dimensional perspective images.

pixel smallest area with unique spatial coordinates (x, y), having a single intensity value or gray level associated with it.

primary visual cortex area of the brain receiving and processing information from the eyes.

resolution a generic term that describes how well a system, process component or image can reproduce an isolated object consisting of separate closely spaced lines or objects.

stereo correspondence problem of determining all pairs of corresponding points from two images of the same scene.

A point p on one image and a point q on a second image are said to form a corresponding pair (p, q) if p and q are each a different sensor projection of the same three dimensional point in the scene.

stereo matching the matching process by which corresponding points on a stereo image pair are identified.

stereopsis the capability of determining the depth of a three dimensional point by observing the point on two perspective projection images taken from different positions.

stereoscope an optical instrument with two eyeglasses for helping the observer to combine the images of two pictures taken from appropriately different points of view to get the effect of solidity or depth.

stereoscopic contour the binocular perception of a well-defined and continuous target boundary which occurs in the absence of an abrupt luminance gradient and for which stereopsis is necessary.