PROCEEDINGS

SPIE—The International Society for Optical Engineering

Intelligent Robots and Computer Vision IX: Neural, Biological, and 3-D Methods

David P. Casasent
Chair/Editor

7–9 November 1990
Boston, Massachusetts

Volume 1382

PROCEEDINGS
SPIE—The International Society for Optical Engineering

Intelligent Robots and Computer Vision IX: Neural, Biological, and 3-D Methods

David P. Casasent
Chair/Editor

7–9 November 1990
Boston, Massachusetts

Sponsored by
SPIE—The International Society for Optical Engineering

Published by
SPIE—The International Society for Optical Engineering
P.O. Box 10, Bellingham, Washington 98227-0010 USA

Volume 1382

The papers appearing in this book comprise the proceedings of the meeting mentioned on the cover and title page. They reflect the authors' opinions and are published as presented and without change, in the interests of timely dissemination. Their inclusion in this publication does not necessarily constitute endorsement by the editors or by SPIE.

Please use the following format to cite material from this book:
Author(s), "Title of Paper," *Intelligent Robots and Computer Vision IX: Neural, Biological, and 3-D Methods,* David P. Casasent, Editor, Proc. SPIE 1382, page numbers (1991).

Library of Congress Catalog Card No. 90-064015
ISBN 0-8194-0449-7

Published by
SPIE—The International Society for Optical Engineering
P.O. Box 10, Bellingham, Washington 98227-0010 USA
Telephone 206/676-3290 (Pacific Time) • Fax 206/647-1445

Printed in the United States of America

INTELLIGENT ROBOTS AND COMPUTER VISION IX:
NEURAL, BIOLOGICAL, AND 3-D METHODS

Volume 1382

CONFERENCE COMMITTEE

Conference Chair

David P. Casasent, Carnegie Mellon University

Program Committee

Mongi A. Abidi, University of Tennessee/Knoxville
Rolf-Jürgen Ahlers, Fraunhofer-Institut für Produktionstechnik und Automatisierung (FRG)
Bruce G. Batchelor, University of Wales College of Cardiff (UK)
Mark F. Cullen, Hughes Danbury Optical Systems, Inc.
Stanley M. Dunn, Rutgers University
Madan M. Gupta, University of Saskatchewan
Ernest L. Hall, University of Cincinnati
James M. Keller, University of Missouri/Columbia
Richard A. Messner, University of New Hampshire
Daniel Raviv, Florida Atlantic University
Steven K. Rogers, Air Force Institute of Technology
Scott A. Starks, University of Texas/El Paso
Matthew A. Turk, Massachusetts Institute of Technology
William E. Wolf, E.I. du Pont de Nemours & Company, Inc.

Session Chairs

Session 1—Neuronal-Morphology of Biological Vision: a Basis for Machine Vision
Madan M. Gupta, University of Saskatchewan

Session 2—Neural Networks for Intelligent Systems
Sanjay R. Natarajan, Carnegie Mellon University

Session 3—Reconstruction, Description, and Modeling of 3-D Surfaces
Daniel Raviv, Florida Atlantic University

Session 4—Three-Dimensional Scene Perception
Mongi A. Abidi, University of Tennessee/Knoxville

Conference 1382, *Intelligent Robots and Computer Vision IX: Neural, Biological, and 3-D Methods*, was part of a three-conference program on Intelligent Robots and Computer Vision held at SPIE's Symposium on Advances in Intelligent Systems, a part of OE/BOSTON '90, 4–9 November 1990, in Boston, Massachusetts. The other conferences were:

Conf. 1381: *Intelligent Robots and Computer Vision IX: Algorithms and Techniques*
Conf. 1383: *Sensor Fusion III: 3-D Perception and Recognition.*

INTELLIGENT ROBOTS AND COMPUTER VISION IX:
NEURAL, BIOLOGICAL, AND 3-D METHODS

Volume 1382

INTRODUCTION

The ninth conference in this series received such a large response that it was separated into two conferences (see Volume 1381). This conference is intended to reflect the newest research results, trends, and developments in intelligent robots and computer vision. The emphasis in this present volume is on neural, biological, and 3-D methods. These three topics comprise respectively one of the three days of this conference. Over 40 papers from seven countries are included, which indicates the international flavor of this conference.

The neural and morphological aspects of machine vision are the first topics considered. These papers emphasize the biological basis and aspects of machine vision. They include new algorithms for a variety of problems in machine vision. Our neural network session continues to attract many of the leading researchers. This year half of the papers were invited ones, most of which are included in the proceedings. The 3-D imaging papers consider reconstruction of 3-D surfaces, techniques to describe and model 3-D surfaces, and methods for 3-D scene perception.

I thank my administrative assistant, Marlene Layton, and my program committee, as well as all the session chairs and authors who made this conference a success and my job more enjoyable.

David Casasent
Carnegie Mellon University

INTELLIGENT ROBOTS AND COMPUTER VISION IX:
NEURAL, BIOLOGICAL, AND 3-D METHODS

Volume 1382

CONTENTS

(continued)

INTELLIGENT ROBOTS AND COMPUTER VISION IX:
NEURAL, BIOLOGICAL, AND 3-D METHODS

Volume 1382

INTELLIGENT ROBOTS AND COMPUTER VISION IX:
NEURAL, BIOLOGICAL, AND 3-D METHODS

Volume 1382

INTELLIGENT ROBOTS AND COMPUTER VISION IX:
NEURAL, BIOLOGICAL, AND 3-D METHODS

Volume 1382

SESSION 1

Neuronal-Morphology of Biological Vision: A Basis for Machine Vision

Chair
Madan M. Gupta
University of Saskatchewan

3-D vision and figure/ground separation by visual cortex

Stephen Grossberg
Boston Univ., Boston Massachusetts 02215

Abstract

A theory of 3-D visual perception and figure/ground separation by visual cortex is described. Called FACADE Theory, it suggests a solution of the 3-D figure/ground problem for biological vision and makes many predictions whereby it can be tested. The theory further develops my 3-D vision theory of 1987, which used multiple receptive field sizes, or scales, to define multiple copies of two interacting systems: a Boundary Contour System (BCS) for generating emergent boundary segmentations of edges, textures, and shading; and a Feature Contour System (FCS) for discounting the illuminant and filling in surface representations of Form-And-Color-And-DEpth, or FACADEs. The 1987 theory did not posit interactions between the several scales of the BCS and FCS. The present theory suggests how competitive and cooperative interactions, that were previously defined within each scale, also act between scales.

Depth determination using Complex Logarithmic Mapping

Sandra L. Bartlett and Ramesh Jain

The University of Michigan, Department of Electrical Engineering and Computer Science
1101 Beal Avenue, Ann Arbor, Michigan 48109

ABSTRACT

There is growing interest in using the complex logarithmic mapping for depth determination in motion stereo applications. This has lead to a need for a comprehensive error analysis. Rather than just giving an analytic description of the errors inherent in the approach, an attempt will be made to characterize the errors that occur when using the mapping with real images. Techniques to reduce the impact of these errors will also be discussed.

1. INTRODUCTION

Living things are very good at seeing. The quality of their high level and low level visual processing would be the envy of any computer vision system being developed today. Studying how living things perform these processes and adapting the methods for machine vision may provide hints for developing effective computer vision systems in the future.

Since much of our visual processing is done in the striate cortex of the brain, it is a good place to look for processing ideas. In many mammals, including man, the mapping between the retina of the eye and the striate cortex can be approximated by a complex logarithmic function [1]. The function itself is interesting because it is a conformal mapping and, under certain conditions, is scale and rotation invariant. Because it possesses these desirable properties and since it is an integral part of biological vision, it seems reasonable to assume that the complex log mapping may prove useful in computer vision as well. Indeed research has shown that it can be used in object recognition [2, 3, 4, 5], computer graphics [6] and motion stereo [7, 8]. The focus of this paper is motion stereo.

Motion stereo capitalizes on the camera motion [9, 10, 11]. If a single camera translates in space, with some displacement in depth, the depths of the stationary objects in the scene can be obtained from their displacements in the image plane and the motion of the camera. One of the problems in motion stereo is that the optical flow vectors point in every direction, while in binocular stereo, the flow is parallel with the epipolar line, which simplifies matching. The complex logarithmic mapping (CLM) is useful for motion stereo because it can be used to orient the flow vectors along image rows so matching is as easy as in binocular stereo. At the same time, using the mapping makes the actual depth computation less complex.

This paper will discuss the accuracy of the depth information that is obtained using CLM for depth determination. First we will define the mapping mathematically and outline its usefulness. Then we will show how it can be used for finding depths. A short discussion of the character of image noise will set the stage for a discussion of some sources of error in the technique. The derivation of this approach is done in the continuous domain, while it is implemented in the discrete image domain. The error this causes is quantified and a suggestion for recovering from it is given. Next image digitization error is discussed, then map digitization error. Again a possible approach for decreasing the impact of these errors is given. Finally there is a description of future work.

2. THE MAPPING

Let us look at the mathematical definition of CLM. The complex log mapping may be written as

$$w = \log z \tag{1}$$

where w and z are complex variables:

$$z = x + iy = r(\cos\theta + i\sin\theta) = re^{i\theta} \tag{2}$$

and

$$w = u(z) + iv(z) \tag{3}$$

In this way, a function or image in z-space with coordinates x and y is mapped to w-space with coordinates u and v. The mapping is obtained from the simplified equations:

$$u(r,\theta) = \log r \tag{4}$$

$$v(r,\theta) = \theta \tag{5}$$

There are many attractive features of this mapping [2, 12, 13]. From the mathematical viewpoint, it is the only analytic function which maps a circular region, such as an image on the retina, into a rectangular region. This is a desirable feature for the study and modeling of the human visual system. The mappings of two regular patterns are shown in Figure 1 to result in similarly regular patterns. Figure 1a shows that concentric circles in an image or in the z-plane become vertical lines in the mapped w-plane. This is because the constant radius, r, at all angles, θ, of the circle gives a constant u coordinate for all v coordinates in the mapped space. Similarly in Figure 1b, an image of radial lines, which have constant angle but variable radii, result in a map of horizontal lines.

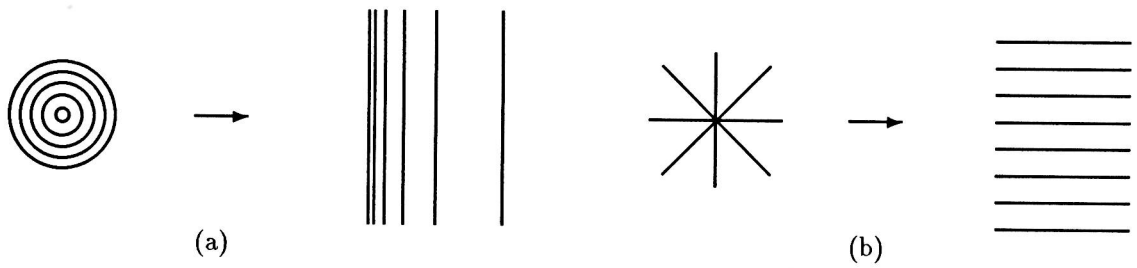

Figure 1: The CLM results in the transformation of certain regular patterns in the z-plane into other regular patterns in the w-plane. (a) shows that concentric circles are mapped into vertical lines. (b) shows that radial lines are mapped into horizontal lines.

Through these mappings, we can demonstrate some interesting properties of CLM. First, it is rotation invariant. In Figure 1a, we saw that for a circle, all possible angular orientations of a point at the given radius will map to the same vertical line. Thus, if an object is rotated between successive images, this will result in only a vertical displacement of the mapped image. This same result can be seen in Figure 1b. As a radial line rotates about the origin, the horizontal line in CLM moves only vertically.

Another property is size invariance. This also can be seen in Figure 1. The mappings of the concentric circles of Figure 1a remain vertical lines and only move horizontally as the circles change in size. As a point moves out from the origin along a radial line in Figure 1b, its mapping stays on the same horizontal line moving only from left to right.

A third important property is projection invariance. When an observer translates in space, the images of stationary objects change on the retina, but the object perceived on the striate cortex does not change. This is due to the fact that in CLM space, translation of the observer only causes the object image to be displaced in the horizontal direction; the size and shape of the object image remain unchanged.

Given these invariences, CLM can be useful for depth perception. Suppose an observer is looking at the center of a circle and moving toward it along his direction of gaze. As he moves, the circle seems to get bigger. If the observer took some pictures of the circle as he moved, they would look much like Figure 1a. As time passes, the vertical line that represents the circle would move in CLM space. How far the circle moves is a function of its depth. Another thing to note is that the size of the circle in CLM space remains the same, even though the size in the image changes, which is very useful for matching.

Unfortunately, these useful properties occur only under certain conditions. Cavanaugh [14, 15] showed that the rotation and scale invariances are obtained only if the object is in the center of the image and the rotation and scale changes are with respect to the origin. The projection invariance is obtained only if the direction of the observer's gaze and motion are the same. However, these constraints are easily satisfied in many computer vision tasks, e.g., a vehicle traveling along a straight road, a space craft docking, or a robot arm approaching the part to be inspected

or grasped. Thus a single operation on the images obtained by a camera mounted on such a moving vehicle or robot can simplify both object recognition and depth determination.

3. DEPTH DETERMINATION

It has been shown [16] that the depth of a static object can be obtained using CLM. Images of the object are obtained by a translating camera. The images are mapped into the CLM space using the focus of expansion (FOE), which can be easily obtained from the motion parameters of the camera, as the origin of the mapping. This is called the ego-motion complex logarithmic mapping (ECLM), since it is based on the motion of the camera itself. Estimates of ego-motion parameters are usually available for moving vehicles and robot arms, so ECLM can be used in these cases. For those applications where the FOE is unknown, there are several techniques available for computing it [7, 17, 18, 19].

If ECLM is used, the depth, Z, of a feature on a static object can be obtained based on the following relation [16]:

$$\frac{du}{dZ} = -\frac{1}{Z} \tag{6}$$

where du is the displacement of the feature along the u axis in the ECLM space and dZ is the distance in depth that the camera moves between frames. Note that

$$\frac{dv}{dZ} = 0 \tag{7}$$

where dv is the displacement along the v axis. This simplification results from the projection invariance of the mapping. Thus the matching features can be found by searching in a line parallel to the axis in ECLM space, rather than searching in a line at an arbitrary angle in the image.

Depth determination using ECLM is similar to binocular stereo in the sense that both methods rely on matching between images - the mapped images for ECLM and the scene images themselves for binocular stereo. ECLM has several advantages, however. First, the search is always along an axis of the image, making it faster and easier to do on the computer. For binocular stereo this is only possible if the two images are obtained from camera positions that are only horizontally or only vertically displaced from each other. Second, the search range for a point in the second image is smaller for ECLM [20]. And finally, the problem of features with no matches is reduced because all the stationary objects in the closer image are also in the farther image.

Using ECLM for real-time depth determination applications requires that the images be mapped into the complex log space in real time. Hardware exists for transforming images at frame rates [21, 22]. CCD sensors which simulate the retino-cortical mapping have been built [23]. Simulations of complex log sensors have also been reported [24, 25].

The error analysis in this paper relates to the following software algorithm for mapping a digital image. For each pixel in ECLM space, the corresponding point in the image is found with subpixel accuracy. The intensity for the ECLM pixel is found by averaging the intensities in the portion of the image covered by a square window centered on the point found from the inverse mapping. Figure 2 shows how this is done. For this work a variable sized window is used to simulate the 50% overlap of the aggregate receptive fields in the retina of the hypercolumns of the primary visual cortex [26]. Also, instead of using uniform weights for the pixels in the window, Gaussian weights are used. While this is not a true model for the receptive fields [27], it represents a compromise between computational complexity and biological fidelity.

Using this approach, the area near the origin of the mapping takes up much more space in the mapped image, while the area far away from the origin takes up less space. Figure 3 shows an example of an image and its CLM. The origin of the mapping is the center of the image, near the woman's nose.

4. IMAGE NOISE

When doing algorithm development, system evaluation, or error analysis for computer vision, most researchers assume that image noise is zero mean white or Gaussian noise with uniform characteristics throughout the image [28, 29, 30, 31, 32, 33]. To test this assumption, 10 images of the same scene were taken, one after the other, with a Javelin Newvichip CCD camera (model JE 7262) and digitized on a Silicon Graphics. Figure 4 shows 2 samples from the set of images. They are indistinguishable to the human eye.

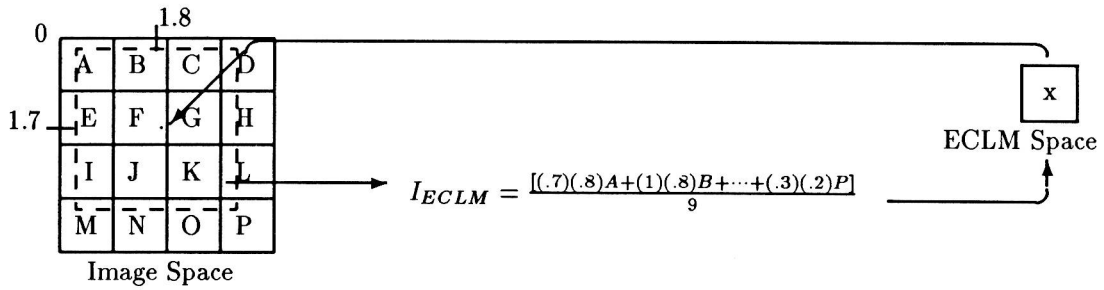

Figure 2: An interpolation scheme used for the ECLM mapping. Note: A,B,...,P are image intensities.

Figure 3: An image and its complex logarithmic mapping.

The variance for each pixel location was calculated and the results are shown in Figure 5. The maximum variance was 10483.440, though the second highest was only 2855.44. The image on the left in the figure shows all the pixels whose variance was 0.0 (there are 19 of them out of 307200). The image on the right, however, is more interesting; it shows all the pixels whose variance was greater than 22. Most of the image acquisition noise is concentrated at locations with maximum intensity gradient, where most image features used in matching are found. If this much variance occurs using the selfsame camera at exactly the same location, even greater differences can be expected using different cameras (binocular stereo) and/or taking the images from different locations (both binocular and motion stereo). This has interesting implications, especially for algorithms, such as ours, that do pixel level matching or comparison. Thus errors in disparity of up to 3 pixels can be routinely expected.

5. CONTINUOUS DERIVATION - DISCRETE IMPLEMENTATION

Equation 6 gives the relationship for finding depth in terms of derivatives. Since the camera cannot move in infinitesimal steps, the equation must be descretized so it can be used in real situations. Thus it becomes

$$\frac{\Delta u}{\Delta Z} = -\frac{1}{Z} \tag{8}$$

where Δu is the difference in u values of the feature in ECLM space in two different frames and ΔZ is the distance the camera moves in depth between the two frames. This is a linear approximation of the differential and gives a depth value relative to the point halfway between the camera location for the first frame and the camera location for the second. Fortunately, this error is easily characterized. Given the exact, subpixel location of matching points in two images obtained from a camera translating in depth, for any given ΔZ, there is a linear relationship between the calculated Δu and the error in the depth calculation. This relationship is constant throughout the image. Figure 6 shows this relationship for some selected values of ΔZ. In the figure, the Δu values represent objects at depths

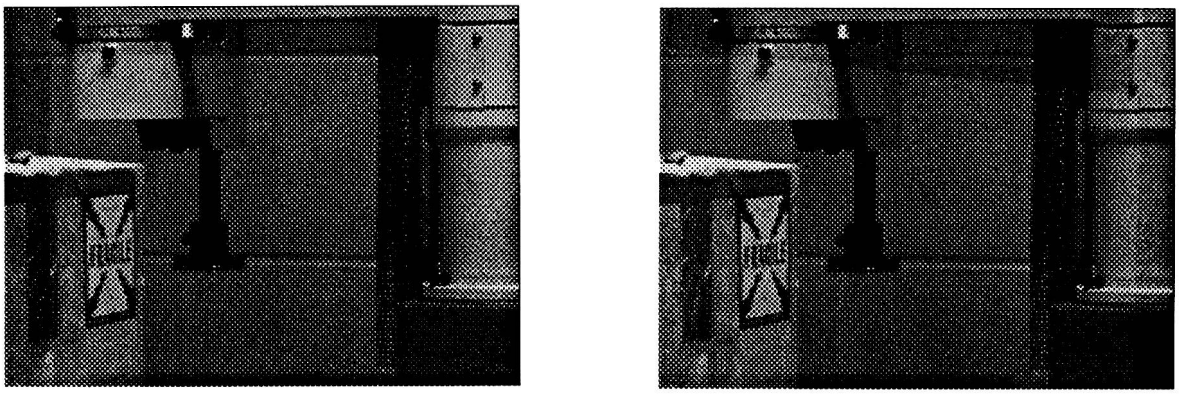

Figure 4: Two samples from the set of test images.

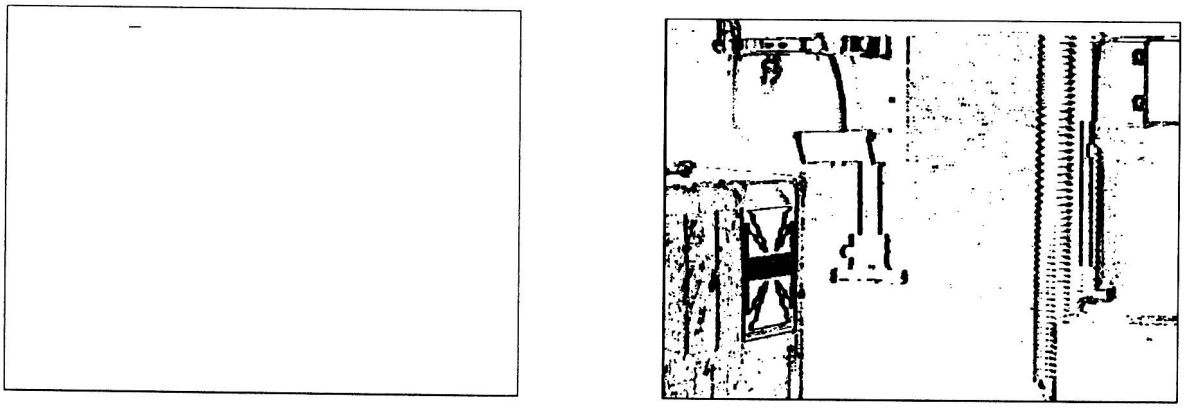

Figure 5: The image on the left shows in black all the pixel locations whose variance was 0.0. The image on the right shows in black all the pixels whose variance was greater than 22 (33120 pixels).

from 25 to 100. (The units are arbitrary, but they must be the same as the ones used for ΔZ.) Note that the errors are smaller for smaller camera movements. Since the correct value of the slope of the error can be easily found for any ΔZ (see Figure 7), this error can be easily and cheaply corrected.

6. IMAGE DIGITIZATION ERROR

In the preceding section it was assumed that the exact, subpixel location of the points of interest were known. Unfortunately, because of the digital nature of the images, this information is not available. If the location of a point in an image is given by (x, y), what this really means is the point may be anywhere in $(x \pm \frac{1}{2}, y \pm \frac{1}{2})$, assuming square pixels. This results in a similar uncertainty in the (u, v) position in ECLM. While the uncertainty in location in the image is constant over the entire image, the resulting uncertainty in the mapping is a function of the distance from the origin of the mapping and the overall effect depends on the size of the image. Figure 8 shows the maximum error in u that can result from image digitization for a 16 x 16 pixel image and a 128 x 128 pixel image. (Note that the error value for the pixel in the center of the image, the origin of the mapping, was chosen arbitrarily for aesthetic reasons; the mapping is, in fact, undefined at that point.)

The error increases significantly the nearer the pixel is to the origin of the mapping (in this case the center of the image). Table 1 shows the data for these size images as well as several others, using the center of the image as the origin of the mapping..

As the image gets larger, the maximum value of u gets larger since the edge of the image is more pixels away from the center. However, the maximum error in u remains constant, because it occurs near the origin. If a 16 x 16 pixel window around the peak in the graph for the 128 x 128 image is compared to the full error graph of the 16 x 16 image, the two would look exactly the same. The error function is constant for all images. The size of the image

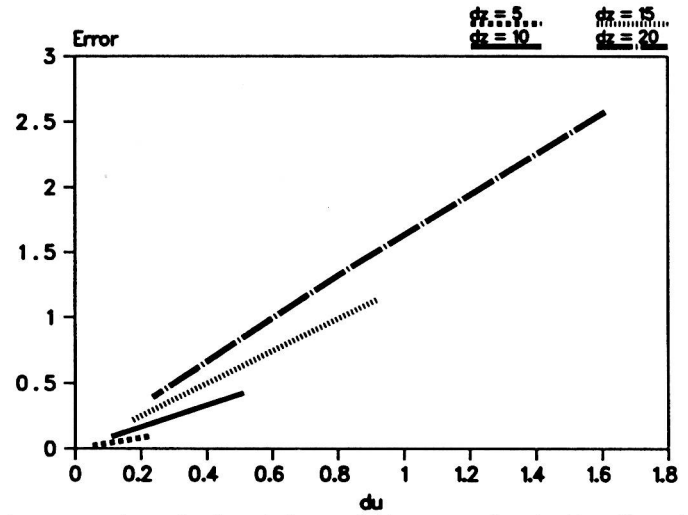

Figure 6: The relationship between the calculated Δu and the error due to the discretization of the differential used to derive the relationship.

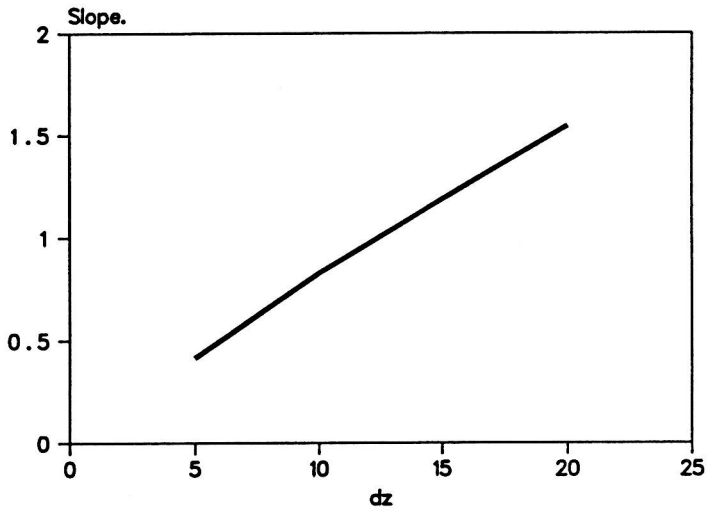

Figure 7: The relationship between the slope of the error curve in the previous figure and ΔZ.

determines how much of the error function takes effect. The farther away from the origin a pixel is located, the less the error. Therefore large images have a smaller minimum error, since the edge of the image is farther away from the origin. Figure 8 and Table 1 show only the error in u. The magnitude of the error in depth depends on how far the camera moves. For any given u error, the absolute depth error will be greater the farther the camera moves.

7. MAPPING DIGITIZATION ERROR

So far the analysis has only considered values of u calculated from image coordinates. These values could be used if feature detection is done in the image. If the image is mapped intc ECLM space and feature detection done there, the digitization of the mapping will also cause errors. As in the image, if the location of a point in the mapping is given by (u, v), what this really means is the point may be anywhere in $(u \pm \frac{1}{2}, v \pm \frac{1}{2})$, assuming square pixels in the mapping. How much error this introduces into the depth computation depends on the size of the mapping, the effect being greater for smaller maps. The size of the image is constrained, to some extent, by the hardware used to obtain it, and is fixed at processing time. When our algorithm is used to map an image, any size of map may be chosen. Given a square image S_i pixels on a side, if the origin of the mapping is the center of the image, the largest

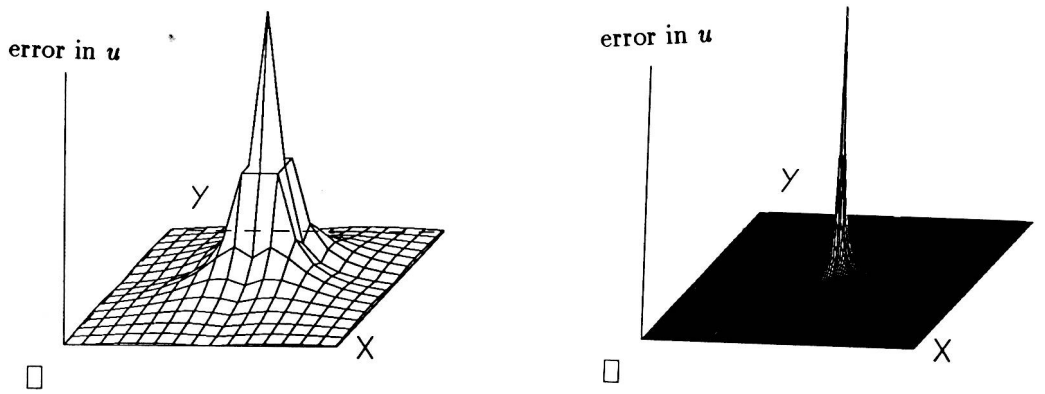

Figure 8: Maximum possible error in u due to image digitization error. The graph on the left is for a 16 x 16 pixel square image. On the right is the graph for a 128 x 128 image.

Image size	Minimum u	Maximum u	Min. u error	Max. u error
16 x 16	0.0	2.426015	0.064539	0.693147
64 x 64	0.0	3.812309	0.015748	0.693147
128 x 128	0.0	4.505457	0.007843	0.693147
256 x 256	0.0	5.198604	0.003914	0.693147
512 x 512	0.0	5.891751	0.001955	0.693147

Table 1: Error in u due to image digitization error for various image sizes.

u value occurs when a corner pixel is mapped. Thus

$$u_{max} = \log \sqrt{\left(\frac{S_i}{2}\right) + \left(\frac{S_i}{2}\right)} \tag{9}$$

If the map is a square image with S_m pixels on a side, then

$$u_{scale} = \frac{u_{max}}{S_m} \tag{10}$$

Table 2 shows u_{scale} values for various sized maps for a 128 x 128 image. Again, the effect on the depth itself depends on how far the camera moves in depth, and is greater for greater camera motion. Table 2 shows that, give

Image size	u in image	Map size				
		16	64	128	256	512
16	2.426015	0.151626	0.037906	0.018953	0.009477	0.004738
64	3.812309	0.238269	0.059567	0.029784	0.014892	0.007446
128	4.505457	0.281591	0.070398	0.035199	0.017599	0.008800
256	5.198604	0.324913	0.081228	0.040614	0.020307	0.010154
512	5.891751	0.368234	0.092059	0.046029	0.023015	0.011507

Table 2: u represented by each map pixel for various map sizes and image sizes.

an image size, you can make the mapping of the u values arbitrarily accurate by choosing a large enough map size. This is another case of trading compute time and memory for accuracy.

Note that u_{scale} defines the depths that can be recovered. The Δu that is required to calculate the depth of a feature is

$$\Delta u = n * u_{scale} \tag{11}$$