

To the Graduate Council:

I am submitting herewith a thesis written by Christophe Delcroix entitled "Analytical Fusion of Edge Maps in Multisensory Images." I have examined the final copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Electrical Engineering.

M. A. Abidi

M. A. Abidi, Major Professor

We have read this thesis
and recommend its acceptance:

M. J. Roberts

R. C. Garzaly

Accepted for the Council:

C. W. Mintel

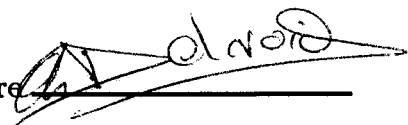
Vice Provost
and Dean of The Graduate School

STATEMENT OF PERMISSION TO USE

In presenting this thesis in partial fulfillment of the requirements for a Master's degree at The University of Tennessee, Knoxville, I agree that the Library shall make it available to borrowers under rules of the Library. Brief quotations from this thesis are allowable without special permission, provided that accurate acknowledgment of the source is made.

Permission for extensive quotation from or reproduction of this thesis may be granted by my major professor, or in his absence, by the Head of Interlibrary Services when, in the opinion of either, the proposed use of the material is for scholarly purposes. Any copying or use of the material in this thesis for financial gain shall not be allowed without my written permission.

Signature

A handwritten signature in dark ink, appearing to be "David", written over a horizontal line.

Date

07 - 29 - 88

ANALYTICAL FUSION OF EDGE MAPS IN MULTISENSORY IMAGES

A Thesis

Presented for the

Master of Science

Degree

The University of Tennessee, Knoxville

Christophe Delcroix

August 1988

To my wife, Maureen.

ACKNOWLEDGEMENTS

The author is grateful to every one who, directly or indirectly, contributed to this research. Dr. M. A. Abidi, in particular, was not only the author's major professor, but also a constant source of support and guidance. The advice and genuine interest of Dr. R. C. Gonzalez and Dr. M. J. Roberts, who served on the committee, are sincerely appreciated.

The author extends his gratitude to Mr. Joel Spears for taking the time to proofread and check the mathematical part of this thesis and for helping in the preparation of its defense, and to Ms. Janet Smith for her much appreciated involvement in the preparation of this thesis.

Finally, the author would like to thank his wife Maureen for her constant support through his studies and also for her invaluable help with English.

This research was sponsored by the Advanced Automation Technology Group, Martin Marietta Aerospace, Denver, Colorado, under contract ORA-E01291-09.

ABSTRACT

In this thesis, a new analytic method for the fusion of range and intensity edge maps is presented. This method focuses on the integration of registered information in order to increase one's confidence in the presence/absence of edges in a depicted scene. The algorithm is based on the interaction between two constraints: the principle of existence, which tends to maximize the value of the output edge map at a given location if one input edge map features a strong edge, and the principle of confirmability, which adjusts this value according to the edge content in the other input edge map at the same location by maximizing the similarity between them. The latter two maximizations are achieved using the Euler-Lagrange Calculus of Variations equations. This method was tested with synthetic range/intensity data. The resulting edge maps combining both range and intensity data satisfy both principles of existence and confirmability.

This method is extended to another type of registered data, namely color edge maps. The range-intensity fusion algorithm is used twice in the frame of its extension to color edge detection. A color component is selected and fused with each of the other two components. This step provides two intermediate edge maps which implicitly contain information about the three color components. These intermediate edge maps are then combined to yield the final color edge map. Experiments were conducted using not only the red, green, and blue representation of color information but also other bases.

TABLE OF CONTENTS

CHAPTER	PAGE
1. Introduction	1
1.1 Statement of the Problem	1
1.2 Related Work	7
1.2.1 Integration of Range and Intensity Data	7
1.2.2 Integration of Color Data	10
1.3 Synopsis	11
2. Fusion of Two Registered Edge Maps	13
2.1 Analytical Formulation	13
2.2 First Order Approximation of the Fusion Functional	17
2.2.1 Fusion Constraints	18
2.2.2 Mathematical Developments	20
2.3 Second Order Approximation of the Fusion Functional	23
2.3.1 Fusion Constraints	24
2.3.2 Mathematical Developments	29
3. Fusion of Three Registered Edge Maps	32
3.1 Simultaneous Merging	32
3.1.1 Analytical Formulation	32

CHAPTER	PAGE
3.1.2 Fusion Constraints	34
3.1.3 Mathematical Developments	37
3.2 Pair-wise Fusion	42
3.2.1 Motivation for this Method	42
3.2.2 Step 1: Pair-wise of Two Input Edge Maps	43
3.2.3 Step 2: Combination of Intermediate Maps	43
4. Numerical Solution	46
4.1 Derivation of the Difference Equation	46
4.2 Direct Solution	51
4.3 Iterative Solution	52
4.3.1 Principle	52
4.3.2 Convergence	53
4.3.3 Implementation	54
5. Experimental Results	57
5.1 Fusion of Range and Intensity Edge Maps	59
5.1.1 Test of Synthetic Data	59
5.1.2 Evaluation of the Fusion Scheme	64
5.2 Fusion of Edge Maps in Color Images	72
5.2.1 Fusion of Color Edge Maps in the RGB Space	74
5.2.2 Fusion of Color Edge Maps in the LT_1T_2 Space	81
5.2.3 Evaluation of the Pair-wise Fusion Scheme	87

CHAPTER	PAGE
6. Conclusions	95
BIBLIOGRAPHY	96
VITA	100

LIST OF FIGURES

FIGURE	PAGE
1.1 Steps involved in the data integration process.	5
2.1 Steps involved in the fusion of two registered images.	14
3.1 Steps involved in the simultaneous fusion of three edge maps. . .	33
3.2 Derivation of the intermediate edge maps.	44
4.1 Mask representing the coefficient of the LHS of the difference equation.	50
5.1 Synthetic intensity image.	60
5.2 Synthetic range image.	61
5.3 Edge map generated from the synthetic intensity image.	62
5.4 Edge map generated from the synthetic range image.	63
5.5 Fusion of range and intensity with $\lambda = 1$: original range edge map (top left), original intensity edge map (top right), fusion function (bottom left), and resulting fused edges (bottom right).	65
5.6 Fusion of range and intensity with $\lambda = 1.5$: original range edge map (top left), original intensity edge map (top right), fusion function (bottom left), and resulting fused edges (bottom right).	66
5.7 Fusion of range and intensity with $\lambda = 2$: original range edge map (top left), original intensity edge map (top right), fusion function (bottom left), and resulting fused edges (bottom right).	67

FIGURE	PAGE
5.8 Fusion of range and intensity with $\lambda = 3$: original range edge map (top left), original intensity edge map (top right), fusion function (bottom left), and resulting fused edges (bottom right).	68
5.9 Fusion of range and intensity with $\lambda = 8$: original range edge map (top left), original intensity edge map (top right), fusion function (bottom left), and resulting fused edges (bottom right).	69
5.10 Thresholded version of the fusion of range and intensity with $\lambda = 3$. The threshold equals 45 on the scale 0–255.	73
5.11 Grape image: red component (top left), green component (top right), and blue component (bottom left).	75
5.12 Grape image: red edge map (top left), green edge map (top right), and blue edge map (bottom left).	76
5.13 Fusion of the red and the green component with $\lambda = .5$: red edge map (top left), green edge map (top right), fusion function (bottom left), and intermediate edge map (bottom right).	78
5.14 Fusion of the red and the blue component with $\lambda = .5$: red edge map (top left), green edge map (top right), fusion function (bottom left), and intermediate edge map (bottom right).	79
5.15 Fusion of the red, green, and blue components with $\lambda = .5$: red edge map (top left), green edge map (top right), blue edge map (bottom left), and fused edge map, which is the result of the combination of intermediate edge maps (bottom right).	80

5.16 Grape image: luminance component (top left), T_1 component (top right), and T_2 component (bottom left).	82
5.17 Grape image: luminance edge map (top left), T_1 edge map (top right), and T_2 edge map (bottom left).	83
5.18 Fusion of the T_1 and the luminance components with $\lambda = .5$: T_1 edge map (top left), luminance edge map (top right), fusion function (bottom left), and fused edge map (bottom right). . . .	84
5.19 Fusion of the T_1 and the T_2 components with $\lambda = .5$: T_1 edge map (top left), T_2 edge map (top right), fusion function (bottom left), and fused edge map (bottom right).	85
5.20 Luminance, T_1 , T_2 , and output edge maps with $\lambda = .5$: luminance edge map (top left), T_1 edge map (top right), T_2 edge map (bottom left), and fused edge map, which is the result of the combination of the intermediate edge maps (bottom right).	86
5.21 Edge map obtained by pair-wise combination of the (red, green) and (red, blue) edge maps with $\lambda = .5$	88
5.22 Edge map resulting from the arithmetic average of the red, green, and blue edge maps.	89
5.23 Edge map derived from the luminance image.	90
5.24 Edge map obtained by pair-wise combination of the (red, green) and (red, blue) edge maps with $\lambda = 2$	91

FIGURE	PAGE
5.25 Edge map obtained by pair-wise combination of the (T_1, L) and the (T_1, T_2) edge maps with $\lambda = .5$	92
5.26 Edge map resulting from the geometric average of the luminance, T_1 , and T_2 edge maps.	94

CHAPTER 1

Introduction

1.1 Statement of the Problem

Computer vision has traditionally utilized an intensity or color camera as a source of information. A camera projects the observed 3-D scene onto a 2-D image plane. The resulting digitized array of data does not explicitly contain the depth component. Scene analysis involves the extraction of various kinds of information from the recorded data in order to describe or build a model of the original scene. In order to perform robotic tasks such as path planning or autonomous guidance, it is highly desirable to obtain information about the geometry of the scene.

Despite their lack of explicit depth information, intensity images contain many clues regarding the shape of objects, and much research has been performed to explore them. Techniques developed for extracting 3-D information from a single intensity image include depth from focusing [1], texture gradient [2], and other sources [3]. Research has also been conducted to obtain range from two or more images. Examples of this approach are stereo imaging [4] and motion related techniques [3]. Contrived lighting approaches have also been

examined [5]. All of the above techniques can be classified as indirect methods. Their performance is limited by several factors including restricting assumptions and/or heavy computational burden [6].

In order to overcome these problems, researchers have explored direct range measuring techniques, which involve producing an energy beam, directing it on the scene, and measuring the reflected signal. Developed measuring systems include triangulation systems, ultrasonic range finders, and laser range finders [7]. The first two of these systems present inherent performance restrictions. The triangulation method suffers from the "missing part" problem [6], which is due to the necessary distance between emitter and receiver. The performance of the ultrasonic range finder is limited because of the tendency of surfaces to reflect an acoustic wave with a reflection angle equal to the incident angle, thus failing to send a strong enough signal back to the receiver when the incident angle is significant. Laser range finders, however, are more accurate. They provide range information either by emitting a laser pulse and measuring the time elapsed between the emission and the detection of the reflected signal, or by emitting a laser beam modulated by a continuous sinusoidal wave and measuring the phase shift between the emitted and the reflected waves. By scanning a scene in a raster fashion, it is possible to record an array of data which contains explicitly the geometry of the scene.

Important research efforts are being devoted to making laser range finders faster and better (improvement of the signal to noise ratio, resolution, etc. . .).

Also, much research is currently being invested in the open field of range data analysis [8]. Due to their explicit 3-D information, range data appear to be more appropriate than intensity data for robotic tasks involving, for instance, autonomous guidance.

However, this recent availability of good quality range images certainly does not render the intensity data obsolete. The question which arises from the simultaneous availability of both sources is how to combine them. This thesis solves some aspects of this problem.

A preliminary obstacle in their combination is the registration problem. It is necessary that the pixel location (i, j) in the intensity and range images correspond to the very same point in the scene. When utilizing a camera and a range finder, which are separated from each other by some distance, the registration task is not a trivial one. However, the use of a laser range finder based on the phase difference measurement solves this problem. The emitted wave modulated laser beam is partially absorbed by the object reflecting it. The level of absorption is related to the reflectance of this object. Consequently, the amplitude of the reflected wave gives a clue to the luminance of the object in the scene. By measuring this amplitude (as well as the phase difference), registered brightness and range data can be obtained. It will be assumed for the remainder of this work that the range and intensity images are registered.

The integration has one of the two following goals: (1) the exploitation of the complementarity of range and intensity in order to obtain hybrid information,

i.e., information which could not be extracted from a single sensory source, or (2) the exploitation of their correlation in order to increase the degree of confidence regarding the presence/absence of a feature. It is clear that, as far as the second case is concerned, correlation is sought, not between the range and intensity data themselves, but between maps of tokens extracted from these data. The feature or token extraction step is necessary in order to present the data in a similar form. It is understood that the selected token will be likely to appear in the two maps.

This thesis presents an analytical method for fusing maps of tokens derived from registered range and intensity images. This principle is illustrated in Fig. 1.1. The goal is to obtain a consistent map of tokens. The token or feature to be selected can be either intermediate forms (edges, shapes, etc...) or structured forms (corners, convex hulls, etc...). However, the selected feature in this study will be limited to edges. The reasons for this choice are both the simplicity of edge detection in intensity and range images, and the wide area of applications requiring edge extraction as a preprocessing step, which this work could effectively improve.

The method presented is based on the interaction of two analytically formulated constraints: the principle of existence, which tends to maximize the value of the output edge map at a given location if the first edge map features an edge at this location, and the principle of confirmability, which modifies this value according to the edge contents in the other edge map at the same location by

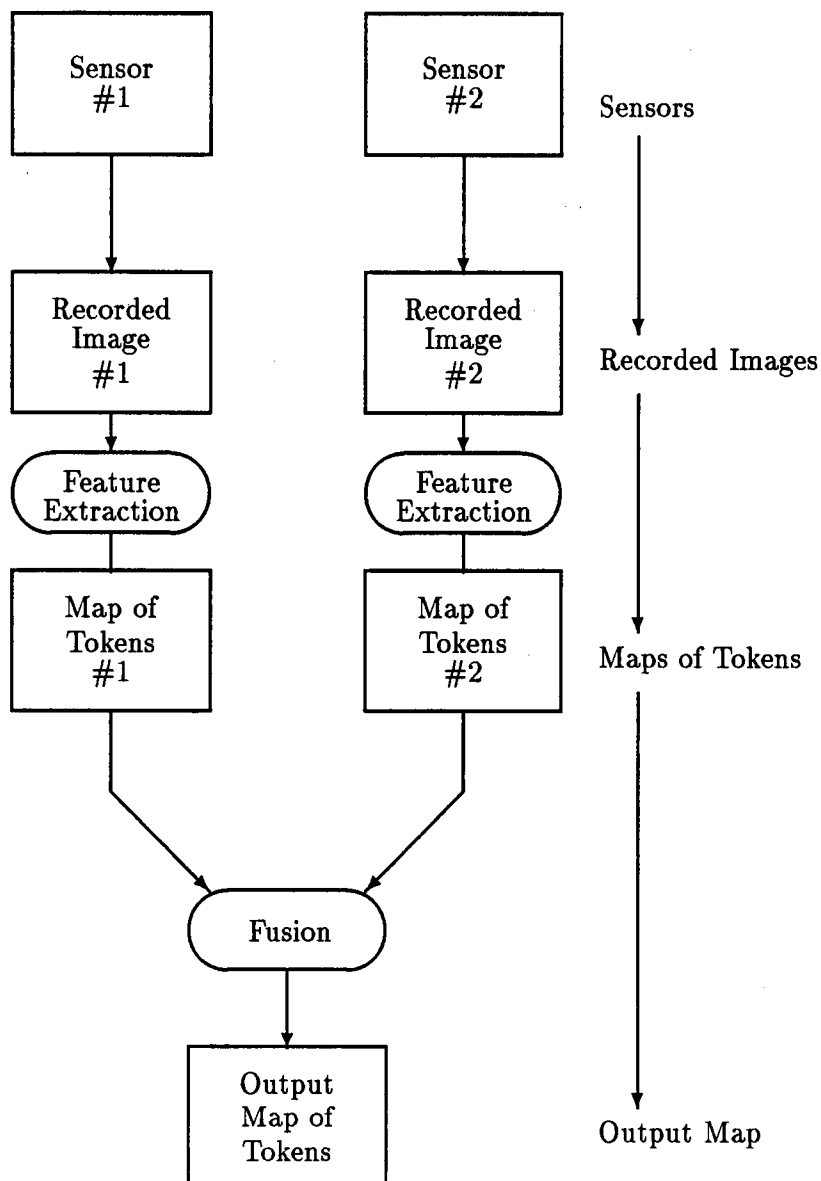


Figure 1.1: Steps involved in the data integration process.

maximizing the similarity between them. The latter two maximization tasks are formulated mathematically as two minimization functions: if the maximal possible edge value is k , maximizing the existence of an edge can be expressed as minimizing the following function: $(k - \text{edge function})$. Similarly, maximizing the similarity between two functions can be expressed as minimizing the difference between them. A new functional called the *fusion functional* is derived by calculating the weighted sum of the latter two minimizing functions: $((\text{existence}) + \lambda^2 \times (\text{confirmability}))$. The minimization of this functional is then achieved using the Euler-Lagrange calculus of variation equations.

The reader should sense the generality of this approach: although it is herein introduced for the integration of range and intensity edge maps, it can easily be extended to other sensory data and different kinds of tokens can be selected. The only two requirements are (1) the registration of the input images and (2) the existence of correlation between the maps of tokens derived from the input images.

The generality of this method will be demonstrated by extending it to the detection of edges in color images. Color information requires a three dimensional basis for its representation. The task to be addressed here is one of calculating independently three edge maps and fusing them. The first approach consists of reformulating the principles of existence and confirmability for the case of three inputs and fusing the three edge maps simultaneously. This approach will be developed theoretically. The other idea involves performing an indirect integra-

tion scheme in which the edge maps are fused by pair (the algorithm performing the fusion of two edge maps needs to be run several times). This method will be implemented and tested.

1.2 Related Work

1.2.1 Integration of Range and Intensity Data

To the best of the author's knowledge, the problem of integrating range and intensity data appeared with the development of the first laser range finders. Garvey [9] describes a system equipped with a color camera and a laser range finder to locate a given object in a scene. The system uses various kinds of knowledge to distinguish the target object from the other ones, and it has the ability to plan each of the steps leading to the object location. This planning, which is based on the optimization of the likelihood of success of the search and the minimization of the time cost, allows the system to sense only the data necessary to complete any particular step. In doing so, it best addresses the problem of sensory data overload. In this work, the data integration is restricted to choosing at each step which one of the range or intensity data should be sensed, acquired, and processed.

Duda *et al.* [10] describe a laser range finder measuring the phase difference between the emitted wave-modulated laser beam and the reflected wave as well as the amplitude of this reflected wave. These two measurements yield registered

range and brightness images of the scene. They show how to use the data in order to perform low level tasks such as the detection of jump boundaries, the extraction of planar surfaces, and the extraction of the normal view of slanted planar surfaces. This last task involves combining range and intensity data in two distinct steps. First, the range data are utilized in order to obtain the location and orientation of an oblique plane. Second, after this plane has been uniformly sampled in its own 2-D cartesian coordinate system, the intensity value is obtained at these points by interpolating among the recorded intensity values. The perspective distortion is thus corrected [11].

In a follow up to this work, using the same laser range finder, Nitzan *et al.* [12] proposed a sequential algorithm to extract planar surfaces of any orientation. First, the major horizontal and vertical surfaces are extracted using exclusively the range data. This step requires that the 3-D cartesian coordinate (x, y, z) be computed for every pixel. This task is performed by applying several transformations to the triplet (i, j, r) where (i, j) denote the image plane coordinate of a pixel and r denotes its associated range value. Major horizontal surfaces are extracted by detecting spikes in the z -histogram, and major vertical surfaces by means of the Hough transform. The other surfaces are then obtained using the intensity data, with the assumption that they have a constant reflectance.

Magee *et al.* [13] present a method for recognizing 3-D objects in multisensory images. The method utilizes both intensity and range data for recognition and avoids the time consuming task of sensing and processing an entire range

image. Their approach consists of recording an intensity image, extracting relevant features from it, and selecting the critical areas for which range information is needed. Depth is then measured for these preselected regions. Aggarwal and Magee pursued this approach of intensity guided range sensing in order to determine motion parameters [14,15].

More recently, Bixler and Miller [16] adopted this idea for robot navigation. They used a low resolution camera in order to obtain a first estimate of the position of an obstacle, and then sense a selected area around the presumed obstacle with a high resolution sonar in order to compute its exact location. They utilized these data to build and update the robot world map for navigational purposes.

Gil *et al.* [17] address the problem of combining intensity and range edge maps. In their study, the Kirsh operator was used for the computation of the intensity gradient, and the range edges were obtained by means of the evaluation of the surface curvature at every pixel location. Two approaches are proposed for the edge map combination. The first one consists of extracting edges appearing in both edge maps by means of a local AND operator. The pixel values from the intensity edge map are AND-ed with the $k \times k$ neighborhood of the corresponding intensity edge map extracted from the range image. The second one is more elaborate. It also consists of extracting common edges, but here, this task is based on a global AND operation. Complete edges are extracted from the intensity image and traced in the range edge map. During this process, short

gaps in the range edge map, if any, can be filled; also, this approach allows for the elimination of spurious short edges by setting a minimal edge length.

1.2.2 Integration of Color Data

Color has not received nearly as much attention as intensity as applied to computer vision. Gershon [21] argues that the major factor was of a technological nature: the memory requirement and the computational burden linked with color processing were prohibitive. However, today's computer power and high memory capabilities have greatly reduced these obstacles. Consequently, color information is being increasingly used for computer vision applications. An example of its use is the CMU Navlab which utilizes a multiclass adaptive color classification scheme for finding and following roads [22].

As far as edge detection in color images is concerned, very little work has been published. Again, this fact contrasts with the significant amount of publications devoted to edge detection in intensity images [23].

Robinson [24] discussed the choice of color coordinates for edge extraction. She considered several bases for the representation of color images (RGB, XYZ, $L^*a^*b^*$, etc...) and derived the corresponding edge maps with the Kirsh operator. The visual differences are evaluated quantitatively by means of two measurements: the energy of the color components and the activity of the edge map associated to each. The final output edge map for each basis is obtained by adding the three corresponding edge maps and thresholding the result.

Nevatia [25] presents an algorithm for color edge detection which represents an extension of the Hueckel intensity edge detector. He adopts an approach to obtain the three color components by which the three edge maps are computed separately, but not quite independently: their computation is linked by a constraint. It is relevant to highlight the principle of the Hueckel edge detector: it consists of fitting an optimal step to the circular neighborhood of a pixel, and of characterizing the actual edge by the parameters of the optimally fitted step function. These parameters are: position, orientation, amplitude, and profile. The algorithm proposed by Nevatia consists of computing the Hueckel edge operator for the three color components with the constraint that, for any given pixel, the direction of the fitted edge be the same for the three images. In other words, one of the four parameters characterizing the edge is not free. The final step consists of the integration of these three edge maps in order to produce the output edge map. This is performed by linking the edges which present compatible color characteristics. Edges are said to be compatible if they have a similar direction when represented in the color space.

1.3 Synopsis

The rest of this thesis is organized as follows. Chapter 2 presents the analytical formulation of the fusion algorithm for two edge maps. Two different orders of precision are investigated. Chapter 3 describes two approaches for the

extention of this algorithm to the fusion of three edge maps. The first approach consists of fusing three edge maps simultaneously; the second one is based on a pair-wise fusion scheme. Chapter 4 deals with the numerical solution. The experiments for the fusion of two and of three edge maps are provided in chapter 5, as well as an evaluation of the performance of the fusion scheme. Finally, some concluding remarks are given in chapter 6.

CHAPTER 2

Fusion of Two Registered Edge Maps

2.1 Analytical Formulation

The task addressed in the frame of this chapter is the fusion of the two edge maps derived from two registered and correlated digital images. The edge values are normalized between zero and one. In order to ease the mathematical formulation, these digital arrays of data are modeled by continuous functions. Let $E_1(x, y)$ and $E_2(x, y)$ denote the two continuous representations of the input edge maps, and $E(x, y)$ denote the output edge map resulting from their fusion. The output can be expressed as an unknown functional F of the two inputs:

$$E(x, y) = F(E_1(x, y), E_2(x, y)).$$

The different steps involved in the data integration process are highlighted in Fig. 2.1. The focus in the following mathematical developments is the determination of the fusion functional F . Assuming that F is analytic, it can be expanded in a converging Taylor series:

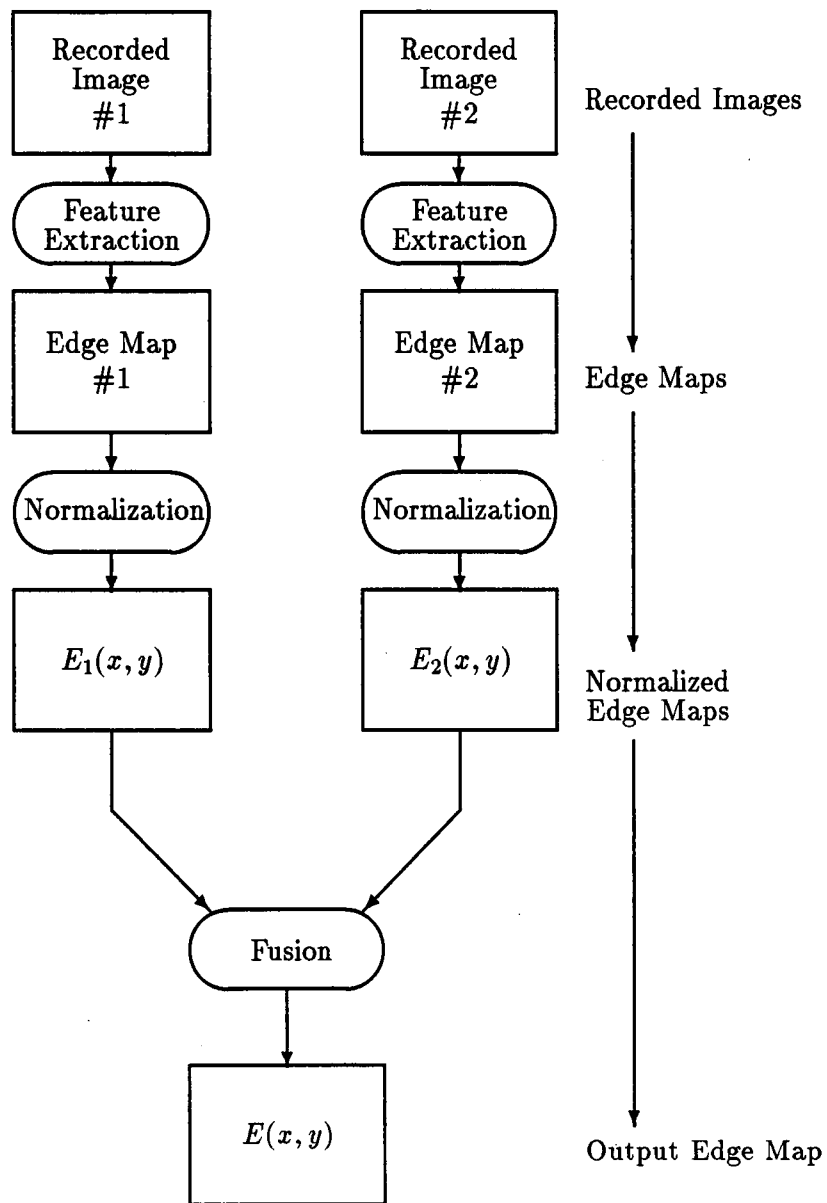


Figure 2.1: Steps involved in the fusion of two registered images.

$$\begin{aligned}
E(x, y) &= F(E_1(x, y), E_2(x, y)) \\
&= F(E_1(x_0, y_0), E_2(x_0, y_0)) \\
&\quad + \left(\frac{\partial}{\partial E_1} F(E_1(x, y), E_2(x, y)) \right) (E_1(x, y) - E_1(x_0, y_0)) \\
&\quad + \left(\frac{\partial}{\partial E_2} F(E_1(x, y), E_2(x, y)) \right) (E_2(x, y) - E_2(x_0, y_0)) \\
&\quad + \frac{1}{2!} \left(\frac{\partial^2}{\partial E_1^2} F(E_1(x, y), E_2(x, y)) \right) (E_1(x, y) - E_1(x_0, y_0))^2 \\
&\quad + \frac{1}{2!} \left(\frac{\partial^2}{\partial E_2^2} F(E_1(x, y), E_2(x, y)) \right) (E_2(x, y) - E_2(x_0, y_0))^2 \\
&\quad + \frac{1}{2!} \left(\frac{\partial^2}{\partial E_1 \partial E_2} F(E_1(x, y), E_2(x, y)) \right) \\
&\quad \times (E_1(x, y) - E_1(x_0, y_0)) (E_2(x, y) - E_2(x_0, y_0)) \\
&\quad + \dots
\end{aligned}$$

The above expression can be developed and then factorized in terms of the powers of the inputs, thereby yielding the formula:

$$E(x, y) = \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} a_{ij}(x, y) E_1^i(x, y) E_2^j(x, y),$$

where $E_1^i(x, y)$ is the i th power of $E_1(x, y)$, $E_2^j(x, y)$ is the j th power of $E_2(x, y)$, and $a_{ij}(x, y)$ are unknown coefficients. Each $a_{ij}(x, y)$ can be expressed as an infinite sum containing partial derivatives of the functional F of the orders $(i+j)$ and higher.

By expanding the functional F in Taylor series, the problem of determining F becomes the problem of determining the unknown coefficients of its Taylor series. However, there are infinitely many a_{ij} 's and it is not possible to determine

accurately each one of them. Consequently, it appears logical to attempt only a good approximation of the real solution. Each coefficient a_{ij} can be written as follows: $a_{ij} = a'_{ij}/(i+j)!$ and it has been proven [26] that the Taylor series expansion of analytic functions converges, *i.e.*, the a'_{ij} 's are bound. Therefore, the value of the a_{ij} 's decreases rapidly with increasing value of $(i+j)$. Similarly, the product $E_1^i(x,y)E_2^j(x,y)$ decreases in this same manner as $E_1(x,y)$ and $E_2(x,y)$ are normalized between "0" and "1". Thus, the contribution of the terms $(a_{ij}(x,y) E_1^i(x,y) E_2^j(x,y))$ decreases significantly as $(i+j)$ increases.

These considerations allow the approximation of the fusion functional F by a truncated version of its Taylor series. What must be determined is the number of terms to be kept. Clearly, the approximation will improve as the order of truncation becomes higher, as more precision is sought. On the other hand, the number of unknown coefficients also increases with the order of truncation. These coefficients can only be determined by means of imposing constraints, and the higher the order of truncation, the more constraints required. The difficulty is that each constraint imposes some form of restriction upon the coefficients. It is imperative that the constraints be meaningful. Otherwise, the restrictions imposed upon the coefficients would be erroneous, and the coefficients would therefore be inaccurate.

Obviously, there must be a compromise between the order of truncation of the series which may yield a good approximation, and the number of coefficients which can be determined with accuracy through meaningful constraints. The

next two sections will investigate two different orders of truncation of the Taylor series.

Note that regardless of the order of truncation of the Taylor series, it appears that one coefficient, namely, $a_{00}(x, y)$, can be discarded ($a_{00}(x, y) \equiv 0$). This coefficient's irrelevance is a consequence of the recognition that all contributions to $E(x, y)$ must be related to the inputs $E_1(x, y)$ and $E_2(x, y)$:

$$E(x, y) = \sum_{\substack{i,j=0 \\ (i,j) \neq (0,0)}}^{\infty} a_{ij}(x, y) E_1^i(x, y) E_2^j(x, y).$$

In the following sections, the dependency upon the independent variables x and y will be assumed, but not explicitly denoted.

$$E = \sum_{\substack{i,j=0 \\ (i,j) \neq (0,0)}}^{\infty} a_{ij} E_1^i E_2^j.$$

2.2 First Order Approximation of the Fusion Functional

In this section, the case will be examined for which the Taylor series expansion of the fusion functional F is truncated at the first order. This rough approximation leads to the following equation:

$$E = a_{10}E_1 + a_{01}E_2, \tag{2.1}$$

which is simply a linear combination of the inputs.

2.2.1 Fusion Constraints

In order to determine the coefficients a_{10} and a_{01} , three meaningful constraints can be expressed.

(A) Normalization Constraint

The expression (2.1) can be normalized so that $0 \leq E \leq 1$. As E_1 and E_2 have already been normalized, this process will affect only the coefficients a_{10} and a_{01} :

$$\begin{aligned} E_{norm} &= \frac{1}{a_{10} + a_{01}} (a_{10}E_1 + a_{01}E_2) \\ &= \alpha E_1 + \beta E_2, \end{aligned}$$

with $\alpha = a_{10}/(a_{10} + a_{01})$ and $\beta = a_{01}/(a_{10} + a_{01})$.

As a result of this scaling, α and β are linked by the relation: $\alpha + \beta = 1$.

For the remainder of this section, the only expression of interest will be the normalized expression. As no ambiguity exists, the reference to the normalization can be omitted:

$$E = \alpha E_1 + \beta E_2.$$

The relation $\alpha + \beta = 1$ reduces the number of unknowns to one.

(B) Principle of Existence

As the goal of this work is to enhance a feature, if any exists, there must be a constraint which allows E to be significant. This constraint is called the principle of existence. As E has been normalized, the highest value it can reach

is one. Thus, the problem of maximizing E can be expressed as the minimization of the expression $(1 - E)$. This constraint can be formulated as follows:

$$\int \int_{scene} (1 - E)^2 dx dy \text{ Minimum.}$$

Let f denote the function $(1 - E)$:

$$f = 1 - E = 1 - (\alpha E_1 + \beta E_2).$$

By using the dependency relation introduced by the normalization constraint,

$$f = 1 - E_2 - (E_1 - E_2)\alpha.$$

(C) Principle of Confirmability

The key to this third constraint is the fact that the two sensors record two fairly correlated and registered arrays of data. Consequently, some similarity is expected between the edge maps E_1 and E_2 . The expectation is that E_1 and E_2 vary at the same location. This property can be termed the principle of confirmability. The presence of an edge in one map should be confirmed by the presence of an edge in the other. Mathematically, the similarity in the variation of the edges can be expressed as a minimization problem of the difference of their derivatives:

$$\int \int_{scene} \|\nabla(\alpha E_1 - \beta E_2)\|^2 dx dy \text{ Minimum,}$$

where the symbol ∇ stands for the differentiation symbol with respect to the two independent variables x and y . Let g denote the minimization function:

$$g = \|\nabla(\alpha E_1 - \beta E_2)\|^2$$

$$\begin{aligned}
&= [(\alpha E_1 - \beta E_2)_x]^2 + [(\alpha E_1 - \beta E_2)_y]^2 \\
&= g_1^2 + g_2^2,
\end{aligned}$$

with

$$\begin{aligned}
g_1 &= (\alpha E_1 - \beta E_2)_x = ((E_1 + E_2)\alpha - E_2)_x \\
&= (E_1 + E_2)\alpha_x + (E_{1x} + E_{2x})\alpha - E_{2x}
\end{aligned}$$

and

$$\begin{aligned}
g_2 &= (\alpha E_1 - \beta E_2)_y \\
&= (E_1 + E_2)\alpha_y + (E_{1y} + E_{2y})\alpha - E_{2y}.
\end{aligned}$$

The notations which were introduced allow the rewriting of the minimization problem as follows:

$$\int \int_{scene} (g_1^2 + g_2^2) dx dy \quad \text{Minimum.}$$

2.2.2 Mathematical Developments

The constraints provide the following equation:

$$\alpha + \beta = 1,$$

and the two following expressions to be minimized simultaneously:

$$\begin{aligned}
&\int \int_{scene} f^2 dx dy, \\
&\int \int_{scene} (g_1^2 + g_2^2) dx dy,
\end{aligned}$$

with

$$\begin{aligned} f &= 1 - E_2 - (E_1 - E_2)\alpha \\ g_1 &= (E_1 + E_2)\alpha_x + (E_{1x} + E_{2x})\alpha - E_{2x} \\ g_2 &= (E_1 + E_2)\alpha_y + (E_{1y} + E_{2y})\alpha - E_{2y} . \end{aligned}$$

In order to optimally minimize the two above integrals simultaneously, it is necessary to create the new function

$$G = f^2 + \lambda^2(g_1^2 + g_2^2),$$

and apply the calculus of variations technique to the integral

$$\int \int_{scene} G \, dx \, dy ,$$

where G explicitly depends upon the two independent variables x and y , and upon $\alpha(x, y)$ and its partial derivatives $\partial\alpha(x, y)/\partial x$ and $\partial\alpha(x, y)/\partial y$. The constant λ must be regarded as a weight factor which allows for the emphasis of either the principle of existence or the principle of confirmability.

If there exists a function $\alpha = \alpha(x, y)$ which minimizes the integral

$$\int \int_{scene} G(x, y, \alpha, \alpha_x, \alpha_y) dx \, dy ,$$

then it must satisfy the Euler-Lagrange (E-L) equation [27,28]:

$$\frac{\partial G}{\partial \alpha} - \frac{\partial}{\partial x} \left(\frac{\partial G}{\partial \alpha_x} \right) - \frac{\partial}{\partial y} \left(\frac{\partial G}{\partial \alpha_y} \right) = 0 .$$

With the following explicit expression for G :

$$\begin{aligned}
G &= f^2 + \lambda^2(g_1^2 + g_2^2) \\
&= (1 - E_2 - (E_1 - E_2)\alpha)^2 \\
&\quad + \lambda^2\{((E_1 + E_2)\alpha_x + (E_{1x} + E_{2x})\alpha - E_{2x})^2 \\
&\quad + ((E_1 + E_2)\alpha_y + (E_{1y} + E_{2y})\alpha - E_{2y})^2\},
\end{aligned}$$

the E-L equation is calculated as detailed here:

$$\begin{aligned}
\frac{\partial G}{\partial \alpha} &= 2(E_2 - E_1)f + \lambda^2\{2(E_{1x} + E_{2x})g_1 + 2(E_{1y} + E_{2y})g_2\} \\
\frac{\partial}{\partial x}\left(\frac{\partial G}{\partial \alpha_x}\right) &= \frac{\partial}{\partial x}(2\lambda^2(E_1 + E_2)g_1) = 2\lambda^2(E_{1x} + E_{2x})g_1 + 2\lambda^2(E_1 + E_2)\frac{\partial g_1}{\partial x} \\
\frac{\partial}{\partial y}\left(\frac{\partial G}{\partial \alpha_y}\right) &= \frac{\partial}{\partial y}(2\lambda^2(E_1 + E_2)g_2) = 2\lambda^2(E_{1y} + E_{2y})g_2 + 2\lambda^2(E_1 + E_2)\frac{\partial g_2}{\partial y}.
\end{aligned}$$

Finally, the E-L equation yields:

$$(E_2 - E_1)f + \lambda^2(E_1 + E_2)\left(\frac{\partial g_1}{\partial x} + \frac{\partial g_2}{\partial y}\right) = 0.$$

By introducing the explicit expression of f , of $\partial g_1/\partial x$, and of $\partial g_2/\partial y$ and by grouping the terms as coefficients of α and its partial derivatives, the following equation results:

$$\begin{aligned}
&\alpha_{xx} + \alpha_{yy} + 2\frac{E_{1x} + E_{2x}}{E_1 + E_2}\alpha_x + 2\frac{E_{1y} + E_{2y}}{E_1 + E_2}\alpha_y \\
&+ \left(\frac{(E_1 - E_2)^2}{\lambda^2(E_1 + E_2)^2} + \frac{\nabla^2(E_1 + E_2)}{(E_1 + E_2)}\right)\alpha = \frac{(1 - E_2)(E_1 - E_2)}{\lambda^2(E_1 + E_2)^2} + \frac{\nabla^2 E_2}{E_1 + E_2}.
\end{aligned}$$

This equation can be written more concisely:

$$\nabla^2 \alpha + \frac{2}{(E_1 + E_2)} < \nabla(E_1 + E_2), \nabla \alpha > + \left(\left(\frac{E_1 - E_2}{\lambda(E_1 + E_2)}\right)^2 + \frac{\nabla^2(E_1 + E_2)}{E_1 + E_2}\right)\alpha = C,$$

with C being the R.H.S. of the previous expression, and $\langle a, b \rangle$ denoting the scalar product of the vector a with the vector b .

It must be noted that the signs of E_{1x} , E_{1y} , E_{2x} , E_{2y} , $\nabla^2 E_1$, and $\nabla^2 E_2$ are not relevant. Only their moduli are meaningful. This statement results from the fact that the sign merely indicates if the transition is from a light to a dark area or vice-versa for the reflectance picture, or from near to far or far to near for the range image. Given the different nature of the data, the kind of transition encountered has no importance. Only the existence and the magnitude of the transition is relevant. Therefore, each derivative and laplacian appearing in the previous equations stands for its own absolute value.

Summary:

The first order approximation of the fusion functional, when modeled continuously, is equivalent to solving one partial differential equation in α of the form:

$$\nabla^2 \alpha + A_1 \alpha_x + A_2 \alpha_y + A_3 \alpha = C.$$

The problem of the existence of a solution will be addressed in the section devoted to the numerical solution.

2.3 Second Order Approximation of the Fusion Functional

This section aims at improving the integration process by truncating the Taylor series expansion of the fusion functional F at a higher order. The second

order truncation yields an expression containing five unknowns. (The constant term has been dropped here, too).

$$E = a_{10}E_1 + a_{01}E_2 + a_{20}E_1^2 + a_{02}E_2^2 + a_{11}E_1E_2. \quad (2.2)$$

2.3.1 Fusion Constraints

The second order approximation differs from the first order approximation by its degree of complexity. However, both approximations address the same problem and, consequently, present similarities. Therefore, some of the constraints expressed below are identical or similar to the ones detailed in section 2.2.1.

(A) Normalization Constraint

The normalization of E is performed by dividing its expression by the sum of the coefficients a_{ij} :

$$\begin{aligned} E_{norm} &= \frac{1}{a_{10} + a_{01} + a_{20} + a_{02} + a_{11}} \\ &\times [a_{10}E_1 + a_{01}E_2 + a_{20}E_1^2 + a_{02}E_2^2 + a_{11}E_1E_2] \\ &= \alpha E_1 + \beta E_2 + \gamma E_1^2 + \delta E_2^2 + \epsilon E_1E_2, \end{aligned}$$

with

$$\begin{aligned} \alpha &= \frac{a_{10}}{a_{10} + a_{01} + a_{20} + a_{02} + a_{11}} \\ \beta &= \frac{a_{01}}{a_{10} + a_{01} + a_{20} + a_{02} + a_{11}} \\ \gamma &= \frac{a_{20}}{a_{10} + a_{01} + a_{20} + a_{02} + a_{11}} \\ \delta &= \frac{a_{02}}{a_{10} + a_{01} + a_{20} + a_{02} + a_{11}} \end{aligned}$$

and

$$\epsilon = \frac{a_{11}}{a_{10} + a_{01} + a_{20} + a_{02} + a_{11}}.$$

The only expression which will be utilized in the remainder of this section is the normalized one. Therefore, as no ambiguity will exist, the reference to the normalization will be omitted:

$$E = \alpha E_1 + \beta E_2 + \gamma E_1^2 + \delta E_2^2 + \epsilon E_1 E_2.$$

The coefficients are linked by the dependency relationship:

$$\alpha + \beta + \gamma + \delta + \epsilon = 1.$$

(B) Principle of Existence

As explained in section 2.2.1, a constraint is needed which maximizes the value of the output E . Maximizing E is equivalent to minimizing the expression $(1 - E)$:

$$\int \int_{scene} (1 - E)^2 dx dy \quad \text{Minimum}$$

Let f denote the minimizing function:

$$f = 1 - E = 1 - (\alpha E_1 + \beta E_2 + \gamma E_1^2 + \delta E_2^2 + \epsilon E_1 E_2).$$

By introducing the dependency relationship between the coefficients,

$$\begin{aligned} f &= 1 - E_1 E_2 - E_1(1 - E_2)\alpha - E_2(1 - E_1)\beta \\ &\quad - E_1(E_1 - E_2)\gamma - E_2(E_2 - E_1)\delta. \end{aligned}$$

(C) Principle of Confirmability

This constraint seeks to emphasize the correlation existing between the two input edge maps: edges are expected to be present in the two maps at the same location, a correlation exploited by the aforementioned principle of confirmability. Within the framework of this second order approximation, the principle of confirmability must be expressed for the first and second order contributions. Indeed, it results from the normalization of the inputs that E_1^2 is smaller than E_1 and E_2^2 is smaller than E_2 . Thus, expressing the principle of confirmability between $(\alpha E_1 + \gamma E_1^2)$ and $(\beta E_2 + \delta E_2^2)$ would systematically tend to act primarily on the larger contributions, i.e., the first order terms. Therefore, confirmability is expressed for both orders:

$$\begin{aligned} \int \int_{scene} ||\nabla(\alpha E_1 - \beta E_2)||^2 dx dy & \text{ Minimum} \\ \int \int_{scene} ||\nabla(\gamma E_1^2 - \delta E_2^2)||^2 dx dy & \text{ Minimum.} \end{aligned}$$

Let g and h denote the minimizing functions:

$$\begin{aligned} g &= ||\nabla(\alpha E_1 - \beta E_2)||^2 \\ &= [(\alpha E_1 - \beta E_2)_x]^2 + [(\alpha E_1 - \beta E_2)_y]^2 \\ &= g_1^2 + g_2^2, \end{aligned}$$

with

$$\begin{aligned} g_1 &= E_1 \alpha_x - E_2 \beta_x + E_{1x} \alpha - E_{2x} \beta \\ g_2 &= E_1 \alpha_y - E_2 \beta_y + E_{1y} \alpha - E_{2y} \beta \end{aligned}$$

and

$$\begin{aligned}
h &= \|\nabla(\gamma E_1^2 - \delta E_2^2)\|^2 \\
&= [(\gamma E_1^2 - \delta E_2^2)_x]^2 + [(\gamma E_1^2 - \delta E_2^2)_y]^2 \\
&= h_1^2 + h_2^2,
\end{aligned}$$

with

$$\begin{aligned}
h_1 &= E_1^2 \gamma_x - E_2^2 \delta_x + 2E_1 E_{1x} \gamma - 2E_2 E_{2x} \delta \\
h_2 &= E_1^2 \gamma_y - E_2^2 \delta_y + 2E_1 E_{1y} \gamma - 2E_2 E_{2y} \delta.
\end{aligned}$$

(D) Optimization of ϵ

The last constraint is more vague than the previous ones and is more of an *ad hoc* type. Let's examine the expression of E and the influence of the constraints (B) and (C) on the coefficients. As E_1 and E_2 are normalized, the constraint (B), which seeks to maximize the numerical value of E , tends to set the coefficient of the largest edge value to one, and the others to zero. The constraint (C) corrects this tendency by stressing the correlation between the input edge maps. However, in neither of these constraints is the value of ϵ emphasized. The coefficient ϵ should follow two tendencies: it must take a significant value when both E_1 and E_2 are significant (and, in addition, should increase and decrease with the value of E_1 and E_2), and it must take a small value at any location where only one map features an edge. This constraint will increase the smoothness of the final edge map.

A straightforward constraint allowing ϵ to be optimized is the following one:

$$\int \int_{scene} (\alpha E_1 + \beta E_2 + \gamma E_1^2 + \delta E_2^2 - u(x, y, E_1, E_2)\epsilon)^2 dx dy \quad \text{Minimum,}$$

where u is a function designed to allow ϵ to vary in the manner described above. Consequently, $u(x, y, E_1, E_2)$ must yield a small value whenever $E_1 \cong E_2$ so that ϵ can reach a significant value (in the interval $[0,1]$), and it must take a substantial value whenever $E_1 \gg E_2$ or $E_2 \gg E_1$. A solution to this problem is $u(x, y, E_1, E_2) = R(E_1, E_2)$, where R denotes the ratio of the highest edge value to the smallest: $R = \max(|E_1/E_2|, |E_2/E_1|)$. Thus, the above constraint becomes

$$\int \int_{scene} (\alpha E_1 + \beta E_2 + \gamma E_1^2 + \delta E_2^2 - R\epsilon)^2 dx dy \quad \text{Minimum.}$$

The requirement that ϵ increase and decrease with the value of E_1 and E_2 also exists in this constraint, as $(R\epsilon)$ tends to reach the value of $(\alpha E_1 + \beta E_2 + \gamma E_1^2 + \delta E_2^2)$ which is proportional to E_1 and E_2 . Let i^2 denote the minimization function:

$$i = \alpha E_1 + \beta E_2 + \gamma E_1^2 + \delta E_2^2 - R\epsilon.$$

By introducing the relation: $\epsilon = 1 - (\alpha + \beta + \gamma + \delta)$,

$$i = (E_1 + R)\alpha + (E_2 + R)\beta + (E_1^2 + R)\gamma + (E_2^2 + R)\delta - R.$$

2.3.2 Mathematical Developments

The constraints provide one dependency relationship:

$$\alpha + \beta + \gamma + \delta + \epsilon = 1$$

and four expressions to be minimized simultaneously:

$$\begin{aligned} & \int \int_{scene} f^2 dx dy \\ & \int \int_{scene} (g_1^2 + g_2^2) dx dy \\ & \int \int_{scene} (h_1^2 + h_2^2) dx dy \\ & \int \int_{scene} i^2 dx dy, \end{aligned}$$

with

$$f = 1 - E$$

$$g_1 = (\alpha E_1 - \beta E_2)_x$$

$$g_2 = (\alpha E_1 - \beta E_2)_y$$

$$h_1 = (\gamma E_1^2 - \delta E_2^2)_x$$

$$h_2 = (\gamma E_1^2 - \delta E_2^2)_y$$

$$i = \alpha E_1 + \beta E_2 + \gamma E_1^2 + \delta E_2^2 - \epsilon R.$$

The dependency relation allows the reduction of the number of unknowns from five to four. By writing $\epsilon = 1 - \alpha - \beta - \gamma - \delta$, the four unknowns to be determined are α , β , γ , and δ .

The approach for simultaneously solving the four minimization tasks is similar to that described in section 2.2.2: it is necessary to create a new minimization function G such that

$$G = f^2 + \lambda^2(g_1^2 + g_2^2) + \mu^2(h_1^2 + h_2^2) + \nu^2 i^2,$$

and to apply the calculus of variations technique. If there exist four functions $\alpha = \alpha(x, y)$, $\beta = \beta(x, y)$, $\gamma = \gamma(x, y)$ and $\delta = \delta(x, y)$ which minimize the integral

$$\int \int_{scene} G(x, y, \alpha, \alpha_x, \alpha_y, \beta, \beta_x, \beta_y, \gamma, \gamma_x, \gamma_y, \delta, \delta_x, \delta_y) dx dy,$$

then they must satisfy the Euler-Lagrange (E-L) equations:

$$\begin{aligned} \frac{\partial G}{\partial \alpha} - \frac{\partial}{\partial x} \left(\frac{\partial G}{\partial \alpha_x} \right) - \frac{\partial}{\partial y} \left(\frac{\partial G}{\partial \alpha_y} \right) &= 0 \\ \frac{\partial G}{\partial \beta} - \frac{\partial}{\partial x} \left(\frac{\partial G}{\partial \beta_x} \right) - \frac{\partial}{\partial y} \left(\frac{\partial G}{\partial \beta_y} \right) &= 0 \\ \frac{\partial G}{\partial \gamma} - \frac{\partial}{\partial x} \left(\frac{\partial G}{\partial \gamma_x} \right) - \frac{\partial}{\partial y} \left(\frac{\partial G}{\partial \gamma_y} \right) &= 0 \\ \frac{\partial G}{\partial \delta} - \frac{\partial}{\partial x} \left(\frac{\partial G}{\partial \delta_x} \right) - \frac{\partial}{\partial y} \left(\frac{\partial G}{\partial \delta_y} \right) &= 0. \end{aligned}$$

Computing the partial derivatives yields the following system:

$$\begin{aligned} (a) \quad & -E_1(1 - E_2)f + \nu^2(E_1 + R)i - \lambda^2 E_1 \left(\frac{\partial g_1}{\partial x} + \frac{\partial g_2}{\partial y} \right) = 0 \\ (b) \quad & -E_2(1 - E_2)f + \nu^2(E_2 + R)i + \lambda^2 E_2 \left(\frac{\partial g_1}{\partial x} + \frac{\partial g_2}{\partial y} \right) = 0 \\ (c) \quad & -E_1(E_1 - E_2)f + \nu^2(E_1^2 + R)i + \mu^2 E_1^2 \left(\frac{\partial h_1}{\partial x} + \frac{\partial h_2}{\partial y} \right) = 0 \\ (c) \quad & -E_2(E_2 - E_1)f + \nu^2(E_2^2 + R)i + \mu^2 E_2^2 \left(\frac{\partial h_1}{\partial x} + \frac{\partial h_2}{\partial y} \right) = 0. \end{aligned}$$

Two linear equations of the type

$$a_1 \alpha + a_2 \beta + a_3 \gamma + a_4 \delta + a_5 = 0$$

can be obtained by calculating the difference

$$\left(\frac{(a)}{E_1} + \frac{(b)}{E_2}\right) \text{ and } \left(\frac{(c)}{E_1^2} + \frac{(d)}{E_2^2}\right).$$

Solving this system of two equations allows the elimination of two unknowns, say γ and δ :

$$\gamma = A_1\alpha + A_2\beta + A_3$$

$$\delta = B_1\alpha + B_2\beta + B_3.$$

By plugging these values into the P.D.E.'s (c) to (d) and again performing the same additions as above, *i.e.*,

$$\left(\frac{(a)}{E_1} + \frac{(b)}{E_2}\right) \text{ and } \left(\frac{(c)}{E_1^2} + \frac{(d)}{E_2^2}\right),$$

two linear relations between α and β are obtained from which α and β can be calculated.

This solution yields the values for α , β , γ , and δ which are obtained regardless of the principle of confirmability. Indeed, the confirmability is implicitly contained in the functions g_1 , g_2 , h_1 , and h_2 which play no role in the determination of the coefficients. This mathematical development shows that, as formulated, this problem is underconstrained. In order to approximate the fusion functional by second order truncation of its Taylor series, the series coefficients need to be further constrained. The generation of additional meaningful constraints is beyond the scope of this thesis. Once these constraints are formulated, the resulting solution is expected to be very complex.

CHAPTER 3

Fusion of Three Registered Edge Maps

3.1 Simultaneous Merging

3.1.1 Analytical Formulation

Let E_1 , E_2 , and E_3 denote the edge maps to be merged. The goal here is to fuse the three edge maps in order to enhance the edge information in the scene. The problem can be illustrated as shown in Fig. 3.1. It can be mathematically modeled in a fashion similar to that adopted for the fusion of two edge maps.

The output E can be expressed as a functional of the three input edge maps:

$$E = F(E_1, E_2, E_3).$$

The functional can be expanded in Taylor series as follows:

$$E = \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} \sum_{k=0}^{\infty} a_{ijk} E_1^i E_2^j E_3^k,$$

where E_m^n denotes the n th power of the m th input, and a_{ijk} the unknown coefficients. The value of the coefficients a_{ijk} decreases sharply with increasing values of i , j and k as their denominator is $(i + j + k)!$, just as the value of $(E_1^i E_2^j E_3^k)$ decreases, since E_1, E_2 , and E_3 are scaled between 0 and 1. Consequently, the contribution of $(a_{ijk} E_1^i E_2^j E_3^k)$ decreases rapidly with increasing

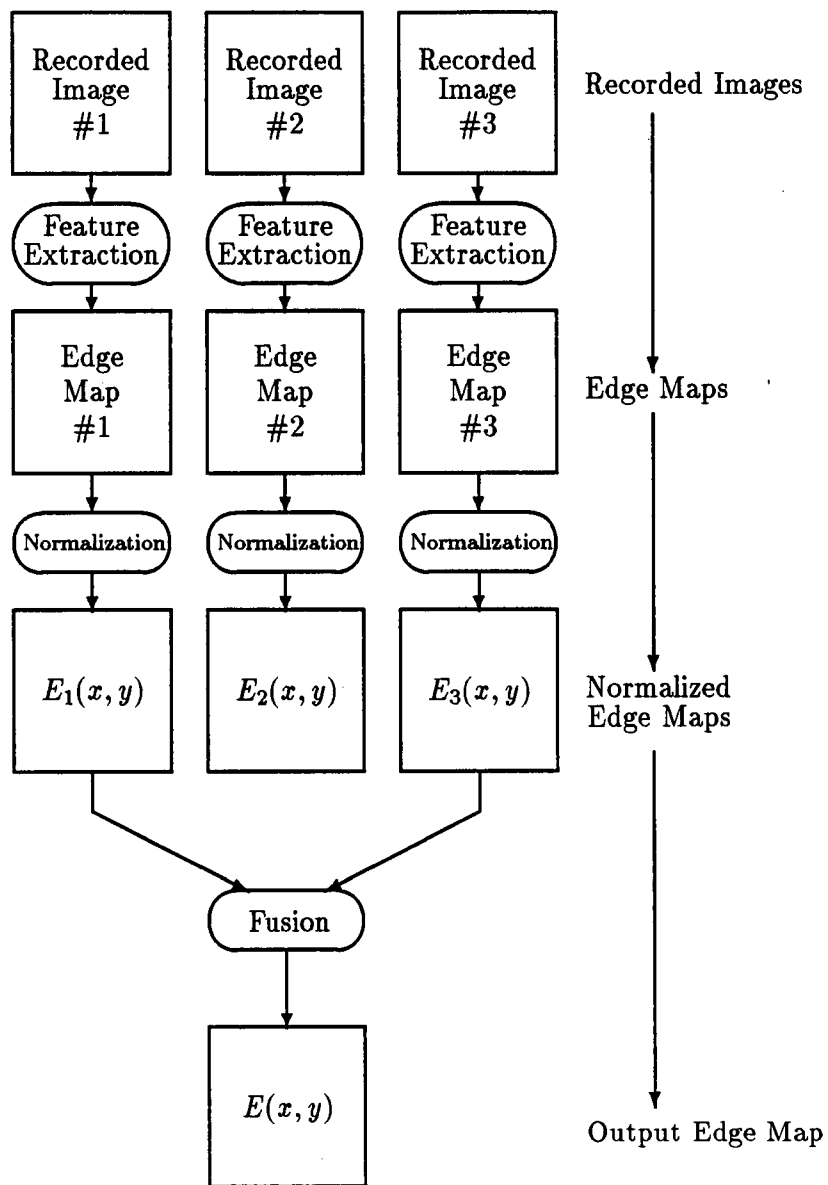


Figure 3.1: Steps involved in the simultaneous fusion of three edge maps.

values of $(i + j + k)$. The fusion functional may be approximated by the first order of its Taylor series expansion:

$$E \cong a_{000} + a_{100}E_1 + a_{010}E_2 + a_{001}E_3.$$

The contribution of higher order terms are considered negligible. In addition, the term a_{000} is set to zero, as all contributions must relate to the inputs. Finally, by renaming the coefficients in the following manner: $a = a_{100}$, $b = a_{010}$, and $c = a_{001}$, the expression of E is as follows:

$$E = aE_1 + bE_2 + cE_3. \quad (3.1)$$

In the following section, the problem of formulating and imposing meaningful constraints for the determination of a , b and c will be addressed.

3.1.2 Fusion Constraints

(A) Normalization Constraint

The normalization of E is obtained by dividing its expression by the sum of the coefficients:

$$\begin{aligned} E_{norm} &= \frac{1}{a + b + c}(aE_1 + bE_2 + cE_3) \\ &= \alpha E_1 + \beta E_2 + \gamma E_3, \end{aligned}$$

with $\alpha = a/(a + b + c)$, $\beta = b/(a + b + c)$, $\gamma = c/(a + b + c)$. The reference to the normalization will be dropped in the remainder of this section:

$$E = \alpha E_1 + \beta E_2 + \gamma E_3.$$

The normalized coefficients are linked by the relationship:

$$\alpha + \beta + \gamma = 1.$$

(B) Principle of Existence

In order to enhance edge information, if any, there must be a constraint allowing E to reach significant values. As E is normalized, the highest value it can reach is one. Thus, the problem of maximizing E can be restated as the task of minimizing $(1 - E)$:

$$\int \int_{scene} (1 - E)^2 dx dy \text{ Minimum.}$$

Let f denote the minimizing function

$$f = 1 - (\alpha E_1 + \beta E_2 + \gamma E_3).$$

By using the relationship among the coefficients, the unknown γ can be replaced by $(1 - \alpha - \beta)$. Thus,

$$f = 1 - E_3 - (E_1 - E_3)\alpha - (E_2 - E_3)\beta.$$

(C) Principle of Confirmability

As the three inputs are highly correlated, the presence of an edge at a pixel location in one map should be confirmed by the presence of an edge at the same location in the other two maps. Consequently, the principle of confirmability can be expressed for every pair of inputs. This statement leads to the minimization of the three following expressions:

$$\int \int_{scene} \|\nabla(\alpha E_1 - \beta E_2)\|^2 dx dy$$

$$\int \int_{scene} ||\nabla(\alpha E_1 - \gamma E_3)||^2 dx dy$$

$$\int \int_{scene} ||\nabla(\beta E_2 - \gamma E_3)||^2 dx dy.$$

For convenience, the following notations will be introduced:

$$\begin{aligned} g &= ||\nabla(\alpha E_1 - \beta E_2)||^2 \\ &= [(\alpha E_1 - \beta E_2)_x]^2 + [(\alpha E_1 - \beta E_2)_y]^2 \\ &= g_1^2 + g_2^2, \end{aligned}$$

with

$$\begin{aligned} g_1 &= E_1 \alpha_x - E_2 \beta_x + E_{1x} \alpha + E_{2x} \beta \\ g_2 &= E_1 \alpha_y - E_2 \beta_y + E_{1y} \alpha + E_{2y} \beta; \end{aligned}$$

$$\begin{aligned} h &= ||\nabla(\alpha E_1 - \gamma E_3)||^2 \\ &= [(\alpha E_1 - (1 - \alpha - \beta) E_3)_x]^2 + [(\alpha E_1 - (1 - \alpha - \beta) E_3)_y]^2 \\ &= h_1^2 + h_2^2, \end{aligned}$$

with

$$\begin{aligned} h_1 &= (E_1 + E_3) \alpha_x + E_3 \beta_x + (E_{1x} + E_{3x}) \alpha + E_{3x} \beta - E_{3x} \\ h_2 &= (E_1 + E_3) \alpha_y + E_3 \beta_y + (E_{1y} + E_{3y}) \alpha + E_{3y} \beta - E_{3y}; \end{aligned}$$

$$\begin{aligned} i &= ||\nabla(\beta E_2 - \gamma E_3)||^2 \\ &= [(\beta E_2 - (1 - \alpha - \beta) E_3)_x]^2 + [(\beta E_2 - (1 - \alpha - \beta) E_3)_y]^2 \\ &= i_1^2 + i_2^2, \end{aligned}$$

with

$$i_1 = E_3\alpha_x + (E_2 + E_3)\beta_x + E_{3x}\alpha + (E_{2x} + E_{3x})\beta - E_{3x}$$

$$i_2 = E_3\alpha_y + (E_2 + E_3)\beta_y + E_{3y}\alpha + (E_{2y} + E_{3y})\beta - E_{3y}.$$

3.1.3 Mathematical Developments

The constraints lead to one linear equation:

$$\alpha + \beta + \gamma = 1$$

and four expressions to minimize:

$$\begin{aligned} & \int \int_{scene} f^2 dx dy \\ & \int \int_{scene} (g_1^2 + g_2^2) dx dy \\ & \int \int_{scene} (h_1^2 + h_2^2) dx dy \\ & \int \int_{scene} (i_1^2 + i_2^2) dx dy, \end{aligned}$$

with

$$f = 1 - E$$

$$g_1 = (\alpha E_1 - \beta E_2)_x$$

$$g_2 = (\alpha E_1 - \beta E_2)_y$$

$$h_1 = (\alpha E_1 - \gamma E_3)_x$$

$$h_2 = (\alpha E_1 - \gamma E_3)_y$$

$$i_1 = (\beta E_2 - \gamma E_3)_x$$

$$i_2 = (\beta E_2 - \gamma E_3)_y.$$

In order to solve simultaneously the four minimization tasks, it is necessary to create a new functional

$$G = f^2 + \lambda^2(g_1^2 + g_2^2) + \mu^2(h_1^2 + h_2^2) + \nu^2(i_1^2 + i_2^2) \quad (3.2)$$

and to find its minimum by means of the calculus of variations. In the equation (3.2), the coefficients λ^2 , μ^2 , and ν^2 allow the emphasis of either of the minimizing functions.

If there exists a pair of functions $\alpha = \alpha(x, y)$ and $\beta = \beta(x, y)$ which minimize the integral

$$\int \int_{scene} G(x, y, \alpha, \alpha_x, \alpha_y, \beta, \beta_x, \beta_y) dx dy,$$

then they must satisfy the Euler-Lagrange (E-L) equations:

$$\begin{cases} \frac{\partial G}{\partial \alpha} - \frac{\partial}{\partial x} \left(\frac{\partial G}{\partial \alpha_x} \right) - \frac{\partial}{\partial y} \left(\frac{\partial G}{\partial \alpha_y} \right) = 0 \\ \frac{\partial G}{\partial \beta} - \frac{\partial}{\partial x} \left(\frac{\partial G}{\partial \beta_x} \right) - \frac{\partial}{\partial y} \left(\frac{\partial G}{\partial \beta_y} \right) = 0. \end{cases}$$

The explicit expression of G is as follows:

$$\begin{aligned} G &= f^2 + \lambda^2(g_1^2 + g_2^2) + \mu^2(h_1^2 + h_2^2) + \nu^2(i_1^2 + i_2^2) \\ &= (1 - E_3 - (E_1 - E_3)\alpha - (E_2 - E_3)\beta)^2 \\ &\quad + \lambda^2 \{ (E_1\alpha_x - E_2\beta_x + E_{1x}\alpha - E_{1x}\beta)^2 \\ &\quad + (E_1\alpha_y - E_2\beta_y + E_{1y}\alpha - E_{2y}\beta)^2 \} \\ &\quad + \mu^2 \{ ((E_1 + E_3)\alpha_x - E_3\beta_x + (E_{1x} + E_{3x})\alpha + E_{3x}\beta - E_{3x})^2 \} \end{aligned}$$

$$\begin{aligned}
& +((E_1 + E_3)\alpha_y + E_3\beta_y + (E_{1y} + E_{3y})\alpha + E_{3y}\beta - E_{3y})^2\} \\
& +\nu^2\{(E_3\alpha_x + (E_2 + E_3)\beta_x + E_{3x}\alpha + (E_{2x} + E_{3x})\beta - E_{3x})^2 \\
& +(E_3\alpha_y + (E_2 + E_3)\beta_y + E_{3y}\alpha + (E_{2y} + E_{3y})\beta - E_{3y})^2\}.
\end{aligned}$$

Detailed computations of the terms in the first E-L equation are as follows:

$$\begin{aligned}
\frac{\partial G}{\partial \alpha} &= -2(E_1 - E_3)f + \lambda^2(2E_{1x}g_1 + 2E_{1y}g_2) \\
&\quad \mu^2(2(E_{1x} + E_{3x})h_1 + 2(E_{1y} + E_{3y})h_2) + \\
&\quad \nu^2(2E_{3x}i_1 + 2E_{3y}i_2) \\
\frac{\partial}{\partial x}\left(\frac{\partial G}{\partial \alpha_x}\right) &= \frac{\partial}{\partial x}(2\lambda^2 E_1 g_1 + 2\mu^2(E_1 + E_3)h_1 + 2\nu^2 E_3 i_1) \\
&= 2\lambda^2 E_{1x}g_1 + 2\lambda^2 E_1 \frac{\partial g_1}{\partial x} + 2\mu^2(E_{1x} + E_{3x})h_1 \\
&\quad + 2\mu^2(E_1 + E_3)\frac{\partial h_1}{\partial x} + 2\nu^2 E_{3x}i_1 + 2\nu^2 E_3 \frac{\partial i_1}{\partial x} \\
\frac{\partial}{\partial y}\left(\frac{\partial G}{\partial \alpha_y}\right) &= \frac{\partial}{\partial y}(2\lambda^2 E_1 g_2 + 2\mu^2(E_1 + E_3)h_2 + 2\nu^2 E_3 i_2) \\
&= 2\lambda^2 E_{1y}g_2 + 2\lambda^2 E_1 \frac{\partial g_2}{\partial y} + 2\mu^2(E_{1y} + E_{3y})h_2 \\
&\quad + 2\mu^2(E_1 + E_3)\frac{\partial h_2}{\partial y} + 2\nu^2 E_{3y}i_2 + 2\nu^2 E_3 \frac{\partial i_2}{\partial y}.
\end{aligned}$$

The first E-L equation is given by:

$$\begin{aligned}
& (E_1 - E_3)f + \lambda^2 E_1 \left(\frac{\partial g_1}{\partial x} + \frac{\partial g_2}{\partial y} \right) + \\
& \mu^2(E_1 + E_3) \left(\frac{\partial h_1}{\partial x} + \frac{\partial h_2}{\partial y} \right) + \nu^2 E_3 \left(\frac{\partial i_1}{\partial x} + \frac{\partial i_2}{\partial y} \right) = 0.
\end{aligned}$$

The second E-L is calculated similarly:

$$-(E_2 - E_3)f - \lambda^2 E_2 \left(\frac{\partial g_1}{\partial x} + \frac{\partial g_2}{\partial y} \right) + \mu^2 E_3 \left(\frac{\partial h_1}{\partial x} + \frac{\partial h_2}{\partial y} \right) +$$

$$\nu^2(E_2 + E_3)\left(\frac{\partial i_1}{\partial x} + \frac{\partial i_2}{\partial y}\right) = 0.$$

The last step toward obtaining the final expression of the E-L equations consists of plugging the explicit expressions of f , g_1 , g_2 , h_1 , h_2 , i_1 , and i_2 into the above two equations. The heavy computations involved in this undertaking will not be included in this report. The result is a system of two partial differential equations which can be written in the following matrix form:

$$\begin{bmatrix} A_1 & A_2 \\ A_3 & A_4 \end{bmatrix} \begin{bmatrix} \nabla^2 \alpha \\ \nabla^2 \beta \end{bmatrix} + \begin{bmatrix} B_1 & B_2 \\ B_3 & B_4 \end{bmatrix} \begin{bmatrix} \alpha_x \\ \beta_x \end{bmatrix} + \begin{bmatrix} C_1 & C_2 \\ C_3 & C_4 \end{bmatrix} \begin{bmatrix} \alpha_y \\ \beta_y \end{bmatrix} + \begin{bmatrix} D_1 & D_2 \\ D_3 & D_4 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} F_1 \\ F_2 \end{bmatrix}.$$

The elements of the first matrix, A_1 , A_2 , A_3 , and A_4 are:

$$\begin{aligned} A_1 &= \lambda^2 E_1^2 + \mu^2 (E_1 + E_3)^2 + \nu^2 E_3^2 \\ A_2 &= \lambda^2 E_1 E_2 + \mu^2 (E_1 + E_3) E_3 + \nu^2 E_3 (E_2 + E_3) \\ A_3 &= \lambda^2 E_1 E_2 + \mu^2 E_3 (E_1 + E_3) + \nu^2 E_3 (E_2 + E_3) \\ A_4 &= \lambda^2 E_2 E_2 + \mu^2 E_3^2 + \nu^2 (E_2 + E_3). \end{aligned}$$

The elements of the second matrix, B_1 , B_2 , B_3 , and B_4 are:

$$\begin{aligned} B_1 &= 2\lambda^2 E_1 E_{1x} + 2\mu^2 (E_1 E_3)(E_{1x} + E_{3x}) + 2\nu^2 E_3 E_{3x} \\ B_2 &= 2\lambda^2 E_1 E_{2x} + 2\mu^2 (E_1 + E_3) E_{3x} + 2\nu^2 E_3 (E_{2x} + E_{3x}) \\ B_3 &= 2(-\lambda^2 E_2 E_{1x} + \mu^2 E_3 (E_{1x} + E_{3x}) + \nu^2 (E_2 + E_3) E_{3x}) \\ B_4 &= 2(\lambda^2 E_2 E_{2x} + \mu^2 E_3 E_{3x} + \nu^2 (E_2 + E_3)(E_{2x} + E_{3x})). \end{aligned}$$

The elements of the third matrix, C_1, C_2, C_3 , and C_4 are:

$$\begin{aligned}
C_1 &= 2\lambda^2 E_1 E_{1y} + 2\mu^2 (E_1 E_3)(E_{1y} + E_{3y}) + 2\nu^2 E_3 E_{3y} \\
C_2 &= 2\lambda^2 E_1 E_{2y} + 2\mu^2 (E_1 + E_3) E_{3y} + 2\nu^2 E_3 (E_{2y} + E_{3y}) \\
C_3 &= 2(-\lambda^2 E_2 E_{1y} + \mu^2 E_3 (E_{1y} + E_{3y}) + \nu^2 (E_2 + E_3) E_{3y}) \\
C_4 &= 2(\lambda^2 E_2 E_{2y} + \mu^2 E_3 E_{3y} + \nu^2 (E_2 + E_3)(E_{2y} + E_{3y})).
\end{aligned}$$

The elements of the fourth matrix, D_1, D_2, D_3 , and D_4 are:

$$\begin{aligned}
D_1 &= \lambda^2 E_1 (E_{1xx} + E_{1yy}) + \mu^2 (E_1 + E_3)(E_{1xx} + E_{1yy} + E_{3xx} + E_{3yy}) \\
&\quad + \nu^2 E_3 (E_{3xx} + E_{3yy}) - (E_1 - E_3)^2 \\
D_2 &= \lambda^2 E_1 (E_{2xx} + E_{1yy}) + \mu^2 (E_1 + E_3)(E_{3xx} + E_{3yy}) \\
&\quad + \nu^2 E_3 (E_{2xx} + E_{2yy} + E_{3xx} + E_{3yy}) - (E_1 - E_3)(E_2 - E_3) \\
D_3 &= \lambda^2 E_2 (E_{1xx} + E_{1yy}) + \mu^2 E_3 (E_{1xx} + E_{1yy} + E_{3xx} + E_{3yy}) \\
&\quad + \nu^2 (E_2 + E_3)(E_{3xx} + E_{3yy}) + (E_2 - E_3)(E_1 - E_3) \\
D_4 &= \lambda^2 E_2 (E_{2xx} + E_{2yy}) + \mu^2 E_3 (E_{3xx} + E_{3yy}) \\
&\quad + \nu^2 (E_2 + E_3)(E_{2xx} + E_{2yy} + E_{3xx} + E_{3yy}).
\end{aligned}$$

Finally, the constants F_1 and F_2 are:

$$\begin{aligned}
F_1 &= (\mu^2 (E_1 + E_3) + \nu^2 E_3)(E_{3xx} + E_{3yy}) - (E_1 - E_3)(1 - E_3) \\
F_2 &= (\mu^2 E_3 + \nu^2 (E_2 + E_3))(E_{3xx} + E_{3yy}) + (E_2 - E_3)(1 - E_3).
\end{aligned}$$

3.2 Pair-wise Fusion

3.2.1 Motivation for this Method

The pair-wise approach to the problem of the merging of three edge maps provides an alternate solution to the one expressed in section 3.1. The motivation for this new method is two-fold. First, it decreases the level of complexity of the three-input problem by transforming it into several two-input problems. Mathematically, this approach changes a complex system of two partial differential equation (P.D.E.) with two unknowns to several independent P.D.E.'s, each of them having only one unknown. This less complicated problem has already been addressed in section 2.2. Secondly, this new method aims at improving the final result. Indeed, there is an inconsistency in the formulation of the simultaneous fusion of three edge maps which concerns the principle of confirmability. It seems that there is no meaningful principle of confirmability integrating the three inputs simultaneously. Therefore, confirmability was sought among pairs of inputs, an approach which appears to be inconsistent with the philosophy of the method. The line of reasoning is as follows: as a constraint cannot be expressed which is appropriate for the method of merging three maps simultaneously, a slightly modified method is introduced so that a full compatibility between the method and the constraints can be reached. This idea leads to the pair-wise integration scheme. It is clear that this approach is not straightforward: after the input edge maps have been fused by pair, another step of

processing is necessary in order to combine these intermediate results in a final output edge map.

3.2.2 Step 1: Pair-wise of Two Input Edge Maps

The input edge maps are fused by pair according to the method given in section 2.2. If the selected pair of input edge maps consists of E_i and E_j , the map resulting from their fusion is denoted by E_{ij} and called "intermediate edge map". It is expected that $E_{ij} = E_{ji}$. The number of inputs allows for three different implementations of this fusion algorithm as shown in Fig. 3.2.

3.2.3 Step 2: Combination of Intermediate Maps

The intermediate edge maps obtained as shown in Fig. 3.2 contain redundant information. Any pair of intermediate edge maps provides indirectly information about the three original inputs. Combining , for example, E_{ij} and E_{ik} means in reality combining a result derived from E_i and E_j with a result derived from E_i and E_k . Consequently, it appears sufficient to utilize only two intermediate edge maps in order to produce the final output map. The efficiency of such an approach must be explored. Utilizing only two intermediate edge maps to produce the output can be considered as privileging one of the three inputs (E_i , for example) fusing it with the other two, and combining the results in order to produce an output. The pair-wise fusion step provides two optimal edge maps with regard to the principles of existence and confirmability, and

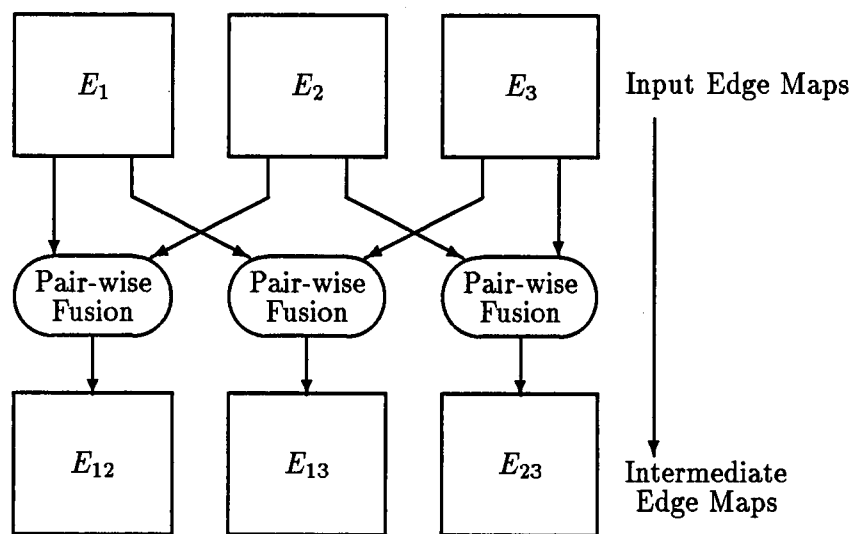


Figure 3.2: Derivation of the intermediate edge maps.

the weight factor λ^2 allows one of the principles to be emphasized. The two intermediate edge maps which are derived as explained can then be combined simply to yield the final edge map. Tests will be conducted using the arithmetic average $(E_{ij} + E_{ik})/2$ and the geometric average $(E_{ij} \cdot E_{ik})^{1/2}$.

CHAPTER 4

Numerical Solution

4.1 Derivation of the Difference Equation

The original problem was the fusion of two discrete edge maps derived from two discrete input images. In order to ease the formulation of the fusion algorithm, these digital arrays were modeled by continuous functions. By approximating the fusion functional by the first order truncation of its Taylor series, the problem of fusing the two continuously modeled edge maps was reduced to one of determining the continuous function $\alpha(x, y)$ in the equation:

$$E(x, y) = \alpha(x, y)E_1(x, y) + (1 - \alpha(x, y))E_2(x, y).$$

It was shown in section 2.2 that, by means of imposing several constraints and applying the calculus of variations technique, $\alpha(x, y)$ must be a solution of the partial differential equation (P.D.E.):

$$\nabla^2 \alpha(x, y) + A_1(x, y)\alpha_x(x, y) +$$

$$A_2(x, y)\alpha_y(x, y) + A_3(x, y)\alpha(x, y) = C(x, y),$$

with

$$A_1(x, y) = 2 \frac{E_{1x}(x, y) + E_{2x}(x, y)}{E_1(x, y) + E_2(x, y)}$$

$$A_2(x, y) = 2 \frac{E_{1y}(x, y) + E_{2y}(x, y)}{E_1(x, y) + E_2(x, y)}$$

$$A_3(x, y) = \left(\frac{E_1(x, y) - E_2(x, y)}{\lambda^2(E_1(x, y) + E_2(x, y))} \right)^2 +$$

$$\frac{\nabla^2 E_1(x, y) + \nabla^2 E_2(x, y)}{E_1(x, y) + E_2(x, y)}$$

and

$$C(x, y) = \frac{(1 - E_2(x, y))(E_1(x, y) - E_2(x, y))}{\lambda^2(E_1(x, y) + E_2(x, y))^2} +$$

$$\frac{\nabla^2 E_2(x, y)}{E_1(x, y) + E_2(x, y)}.$$

After having transformed the original discrete framework to the continuous domain for the convenience of the formulation, it is now necessary to return to the discrete framework in order to solve the original task. The new notation, $\alpha(i, j)$, as well as $A_1(i, j)$, $A_2(i, j)$, $A_3(i, j)$, and $C(i, j)$, will be introduced in order to emphasize the discrete character of the next equations. The transformation from the continuous to the discrete model is obtained by means of the

following approximations:

$$\begin{aligned}
\alpha(x, y) &= \alpha(i, j) \\
\alpha_x(x, y) &= (\alpha(i+1, j) - \alpha(i-1, j))/2 \\
\alpha_y(x, y) &= (\alpha(i, j+1) - \alpha(i, j-1))/2 \\
\nabla^2 \alpha(x, y) &= \alpha(i+1, j) + \alpha(i-1, j) + \\
&\quad \alpha(i, j+1) + \alpha(i, j-1) - 4\alpha(i, j) \\
&= \nabla^2 \alpha(i, j).
\end{aligned}$$

By introducing these discrete approximations, the continuous partial differential equation becomes a difference equation, whose expression is given here:

$$\begin{aligned}
&(A_3(i, j) - 4)\alpha(i, j) + (1 - \frac{A_1(i, j)}{2})\alpha(i-1, j) + (1 - \frac{A_2(i, j)}{2})\alpha(i, j-1) \\
&\quad + (1 + \frac{A_2(i, j)}{2})\alpha(i, j+1) + (1 + \frac{A_1(i, j)}{2})\alpha(i+1, j) = C(i, j).
\end{aligned}$$

The coefficients are also computed discretely, and the derivatives and laplacians which appear in their expressions are approximated in the very same manner as the derivatives and laplacians of $\alpha(x, y)$:

$$A_1(i, j) = \frac{|E_1(i+1, j) - E_1(i-1, j)| + |E_2(i+1, j) - E_2(i-1, j)|}{E_1(i, j) + E_2(i, j)}$$

$$A_2(i, j) = \frac{|E_1(i, j+1) - E_1(i, j-1)| + |E_2(i, j+1) - E_2(i, j-1)|}{E_1(i, j) + E_2(i, j)}$$

$$A_3(i, j) = \left(\frac{E_1(i, j) - E_2(i, j)}{\lambda(E_1(i, j) + E_2(i, j))} \right)^2 + \frac{|\nabla^2 E_1(i, j)| + |\nabla^2 E_2(i, j)|}{E_1(i, j) + E_2(i, j)}$$

$$C(i, j) = \frac{(1 - E_2(i, j))(E_1(i, j) - E_2(i, j))}{\lambda^2(E_1(i, j) + E_2(i, j))^2} + \frac{|\nabla^2 E_2(i, j)|}{E_1(i, j) + E_2(i, j)}.$$

The absolute values were introduced for the computation of the derivatives and laplacians of E_1 and E_2 , as a consequence of the irrelevance of their sign as explained at the end of the section 2.2. For conciseness, the coefficients of the difference equation will be written as follows:

$$\begin{aligned} T_{ij} &= 1 - \frac{A_1(i,j)}{2} \\ B_{ij} &= 1 + \frac{A_1(i,j)}{2} \\ L_{ij} &= 1 - \frac{A_2(i,j)}{2} \\ R_{ij} &= 1 + \frac{A_2(i,j)}{2} \\ M_{ij} &= A_3(i,j) - 4 \end{aligned}$$

and

$$C_{ij} = C(i,j).$$

Thus, the difference equation becomes

$$\begin{aligned} T_{ij}\alpha(i-1,j) + L_{ij}\alpha(i,j-1) + M_{ij}\alpha(i,j) + R_{ij}\alpha(i,j+1) + B_{ij}\alpha(i+1,j) \\ = C_{ij}. \end{aligned} \tag{4.1}$$

The coefficients of the difference equation can be expressed as a pixel dependent mask [Fig. 4.1]. This mask, applied at every possible picture element of the two dimensional array of $\alpha(i,j)$, yields the LHS of Eqn. 4.1. Note that this mask is applicable at every pixel location with the exception of the ones situated on the boundaries. Therefore, only $(N-2)^2$ equations can be written, which

	T_{ij}	
L_{ij}	M_{ij}	R_{ij}
	B_{ij}	

Figure 4.1: Mask representing the coefficient of the LHS of the difference equation.

involve $(N - 2)^2 + 4(N - 2)$ unknowns. There are obviously more unknowns than equations. In order to render this problem solvable, it is necessary to guess the values of $\alpha(i, j)$ on the boundary, either by assignment to a fixed value, or by extrapolation. This process transforms the original unbounded problem to a boundary-valued problem. The task, then, becomes one of solving a system of $(N - 2)^2$ equations with $(N - 2)^2$ unknowns.

The resulting system can be written in a matrix form as follows:

$$A\alpha = C. \quad (4.2)$$

In this equation, the vector α has the dimension $(N - 2)^2$ and was derived from the original two dimensional array of $\alpha(i, j)$'s by reordering it according to the following scheme:

$$\alpha(p) = \alpha(iN + j) = \alpha(i, j),$$

for $1 \leq i \leq (N - 2)$ and $1 \leq j \leq (N - 2)$. The matrix A is a $(N - 2)^2 \times (N - 2)^2$ matrix and contains the coefficients T_{ij} , L_{ij} , M_{ij} , R_{ij} , and B_{ij} which are positioned according to the one dimensional reordering of the α 's. The vector C is of

size $(N - 2)^2$. With the particular reordering of the α 's, the $(N - 2)^2 \times (N - 2)^2$ matrix A takes a simple form. The matrix A can be subdivided into $(N - 2)^2$ submatrices of size $(N - 2) \times (N - 2)$. Let $\mathcal{A}_{ij}$ denote these submatrices for $1 \leq i, j \leq (N - 2)$. Any matrix $\mathcal{A}_{ij}$ can be regarded as an $(N - 2) \times (N - 2)$ element of the matrix A . For any given i , all matrices $\mathcal{A}_{ij}$ are identical to zero with the exception of $\mathcal{A}_{i,i-1}$ (if $i > 1$), $\mathcal{A}_{ii}$, and $\mathcal{A}_{i,i+1}$ (if $i < N - 2$). Furthermore, $\mathcal{A}_{i,i-1}$ and $\mathcal{A}_{i,i+1}$ are diagonal. The only elements of $\mathcal{A}_{ii}$ which can be different from zero are the ones situated on the diagonal, upper diagonal, or lower diagonal. Consequently, the matrix A is a sparse matrix. At most, five of its coefficients are different from zero per line and per column, and their location is known. The next sections will address the task of solving the Eqn. 4.2.

4.2 Direct Solution

The direct solution is straightforward and consists of inverting the matrix A in order to obtain the unknown vector $\underline{\alpha}$:

$$\underline{\alpha} = A^{-1}C.$$

Given the large size of the matrix A , the direct solution can only be implemented if A presents some nice properties which allow the use of an algorithm, which avoids most of the computational burden linked with the classical matrix inversion method. Matrices easily invertible include symmetrical, block-symmetrical, block-diagonal matrices, etc. . . . Other particular properties have also been ex-

plored [29]. Despite the fact that A is a sparse matrix and that the location of its non-zero coefficients follows an apparently simple pattern, the matrix A does not belong to any of the categories already explored. The large number of its coefficients equal to zero would certainly allow for a reduction of the computational burden in the inversion process, but this task would still be time consuming. This approach will not be pursued further.

4.3 Iterative Solution

4.3.1 Principle

The purpose of iterative methods is to reach or approximate the exact solution by a sequence of vectors $\underline{\alpha}^{(1)}, \underline{\alpha}^{(2)}, \dots, \underline{\alpha}^{(k)}, \dots$ constructed according to the following recurrence formula [30]:

$$\underline{\alpha}^{(k)} = \underline{\alpha}^{(k-1)} + H^{(k)}(C - A\underline{\alpha}^{(k-1)}) \quad (4.3)$$

or

$$\underline{\alpha}^{(k)} = (I - H^{(k)}A)\underline{\alpha}^{(k-1)} + H^{(k)}C, \quad (4.4)$$

where $H^{(1)}, H^{(2)}, \dots, H^{(k)}, \dots$ is a sequence of matrices, the choice of which will characterize the iteration process. For instance, the classical process of successive approximation is reached if $H^{(k)}$ is chosen to be the identity matrix ($H^{(k)} = I$). The stationary process corresponds to the case for which the matrix H^k is fixed for all steps ($H^{(k)} = H$).

The above iteration formulas are obtained by multiplying the original system by a matrix H before solving it:

$$HA\underline{\alpha} = HC$$

or

$$(HA - I + I)\underline{\alpha} = HC$$

or

$$\underline{\alpha} = (I - HA)\underline{\alpha} + HC$$

or

$$\underline{\alpha} = \underline{\alpha} + H(C - A\underline{\alpha}).$$

The role of the matrix H will be explained in the next paragraph. By selecting a different matrix $H^{(k)}$ at every step, the Eqn. 4.3 and 4.4 are immediately retrieved from the above derivation.

4.3.2 Convergence

Let $\underline{\alpha}^*$ denote the exact solution. The vector $\underline{\alpha}^*$ fulfills the equation: $A\underline{\alpha}^* = C$. At a given step $(k-1)$, the algorithm yields the approximation $\underline{\alpha}^{(k-1)}$ of $\underline{\alpha}^*$. The residual vector, which is a measure of the difference between the approximate solution and the exact one, amounts to $(C - A\underline{\alpha}^{(k-1)})$. It can be seen from the Eqn. 4.3 that $\underline{\alpha}^{(k-1)}$ is modified by a quantity proportional to its residual vector to yield $\underline{\alpha}^{(k)}$. The success of the iterative process, *i.e.*, the guarantee that the sequence $\underline{\alpha}^{(1)}, \underline{\alpha}^{(2)}, \dots, \underline{\alpha}^{(p)}, \dots$ converges toward $\underline{\alpha}^*$, requires that the residual

vector converge toward zero. It has been shown [29] that the convergence requirement will be met if and only if the eigenvalues of the matrix $(I - H^{(k)}A)$ in the Eqn. 4.4 are smaller than one in modulus. It is clear now that the matrix H can be chosen at every step k in order to fulfill the above convergence condition. In fact, the matrix H can have two purposes: either to make the method converge or to increase its speed of convergence. The “best” matrix $H^{(k)}$ would be such that, at every step k , the eigenvalues of $(I - H^{(k)}A)$ are smaller than but close to one in modulus. However, this determination is itself time costly. It has been shown [29] that a sufficient condition assuring that the eigenvalues of a given matrix are smaller than one is that any norm of this matrix is smaller than one. This criterion will be utilized. The matrix H will be equal to hI , where h is a constant and I stands for the $(N - 2)^2 \times (N - 2)^2$ identity matrix. The constant h is proportional to the inverse of the norm of the matrix A . If the elements of the matrix A are denoted by a_{ij} , the norm is calculated as follows:

$$||A|| = \min_i \left(\sum_{j=1}^{N-2} |a_{ij}| \right),$$

for $i = 1, 2, \dots, N - 2$. It is clear that the selected iterative process is of a stationary nature.

4.3.3 Implementation

From a programming point of view, it is not possible to implement the iteration process by means of matrices according to the general scheme:

$$\underline{\alpha}^{(k)} = D\underline{\alpha}^{(k-1)} + E,$$

with $D = (I - HA)$ and $E = HC$. Indeed, the size of the matrix D is prohibitive. For example, if the resolution of the recorded arrays of sensory data equals 128×128 , the number of unknowns, after fixing the boundary values, equals $(126)^2$. Thus, the matrix D would be of size $126^2 \times 126^2$. As the elements of D are floating point numbers, its declaration would require about 32 Mbytes of memory. This size is too large, even for a main frame computer. However, the reduced number of coefficients different from zero in the matrix $D = (I - HA)$ allows for an approach requiring minimal memory space. Instead of storing the coefficients in large matrices, the original form of the difference equation will be kept:

$$\begin{aligned}\alpha^{(k)}(i, j) = & -hT_{ij}\alpha^{(k-1)}(i-1, j) - hL_{ij}\alpha^{(k-1)}(i, j-1) + \\ & (1 - hM_{i,j})\alpha^{(k-1)}(i, j) \\ & -hR_{ij}\alpha^{(k-1)}(i, j+1) - hB_{ij}\alpha^{(k-1)}(i+1, j) + hC_{ij}.\end{aligned}$$

Writing this equation for all i 's and j 's, with $1 \leq i, j \leq (N - 2)$, implements the iterative process with minimal memory requirement. As each of the $(N - 2)^2$ equations have at most five coefficients, the memory allocation required of this implementation amounts to less than $5(N - 2)^2 \times 4$ bytes, i.e., with $N = 128$, 320 Kbytes, as compared with the 32 Mbytes for the matrix formulation.

This method can be slightly modified in order to yield a cyclic process, by which all the available $\alpha^{(k)}(m, n)$ are utilized when computing $\alpha^{(k)}(i, j)$:

$$\alpha^{(k)}(i, j) = -hT_{ij}\alpha^{(k)}(i-1, j) - hL_{ij}\alpha^{(k)}(i, j-1) +$$

$$(1 - hM_{i,j})\alpha^{(k-1)}(i, j) \\ - hR_{ij}\alpha^{(k-1)}(i, j + 1) - hB_{ij}\alpha^{(k-1)}(i + 1, j) + hC_{ij}.$$

This modification presents the advantages of always utilizing the latest information and of further reducing the memory requirement, as the vector $\underline{\alpha}$ is cyclically updated (there is no need for having $\underline{\alpha}^{(k-1)}$ memorized while $\underline{\alpha}^{(k)}$ is being computed).

CHAPTER 5

Experimental Results

This chapter illustrates by means of synthetic and real data the previously described method for edge fusion. The input data consist of range and intensity images for the fusion of two edge maps and of color images for the pair-wise fusion of three edge maps.

As explained in the previous chapters, this study focuses on the fusion process, not on any of the preprocessing steps from the recorded data to the edge maps. The operator selected for deriving the edges is the Sobel operator [31]. The Sobel edge detector measures the discontinuity in the data at every pixel location by approximating its x and y first derivatives. It is widely utilized for intensity images. However, it is not completely suited for range data. Surfaces which are slanted with regard to the image plane present a strong first derivative, while the Sobel operator might yield a very low response at the intersection of slanted surfaces. For these two cases, the Sobel operator will produce an unsatisfactory value. However, the Sobel operator performs well at jump boundaries. Its use is appropriate for noiseless synthetic images containing few edges.

It is clear that some areas in both images do not contain edges. In order to avoid running the edge fusion algorithm for these areas, a threshold on the edge

intensity must be specified. The edges will be fused only if their sum exceeds this threshold. Also, a value for λ must be specified. The mathematical expression which is minimized is $F(\alpha) = (\text{existence}(\alpha) + \lambda \text{ confirmability}(\alpha))$. As we have no knowledge of how to choose or constrain λ , it is left as a parameter. As mentioned in the theoretical part, λ could be constrained so that its optimal value can be derived, but addressing this issue goes beyond the scope of this study. Tests are conducted with several values of λ . These values can be chosen so that both constraints on α have a sensible influence on its computation, or so that one constraint be privileged.

The α 's obtained should theoretically fall in the interval zero to one. However, because of numerical errors, some values of α are negative, and some are greater than one. This result can be explained by the fact that the edge maps' second derivatives are required in the computation of the α 's, and the high noise sensitivity of the second derivatives yield erroneous values which affect the value of the α 's. A pure rescaling between zero and one would compress the sensible values of α in a narrow interval. In order to eliminate the noise, the α 's must be multiplied by a factor and then thresholded.

The results are presented as follows: The input edge maps to be fused appear in the top left and top right corners of the pictures. The map containing the values of α (scaled between 0 and 255), which is referred to as the fusion function, appears in the bottom left corner, and the result $(\alpha \times \text{Input1} + (1 - \alpha) \times \text{Input2})$ is displayed in the bottom right corner.

5.1 Fusion of Range and Intensity Edge Maps

5.1.1 Test of Synthetic Data

The synthetically created scene consists of a cube on a vertical uniform background. The background is white, and the cube has been painted in a graded fashion. Its front side is black, and the two visible lateral sides are black from their intersection with the front side to their middle part before gradually changing to lighter shades. Their brightness at their intersection with the back side of the cube is similar to the brightness of the background. In addition, a light label appears on the cube's dark front side. The intensity and range pictures are displayed in Fig. 5.1 and Fig. 5.2. Their corresponding scaled edge maps are displayed in Fig. 5.3 and Fig. 5.4.

There exists both differences and similarities in the edge map derived from the range image and the edge map derived from the intensity image. The label which appears on the front side of the cube in the intensity edge map is totally absent in the range edge map. The edges located at the intersection of the back side of the cube with the background are present in both images, but appear much stronger in the range edge map than in the intensity edge map. Finally, the edges located at the intersection of the front side of the cube with the background are strong in both edge maps. Due to the use of the Sobel operator, weak edges appear in both maps all over the lateral sides of the cube. They correspond to the gradual changes either in the range or in the intensity values

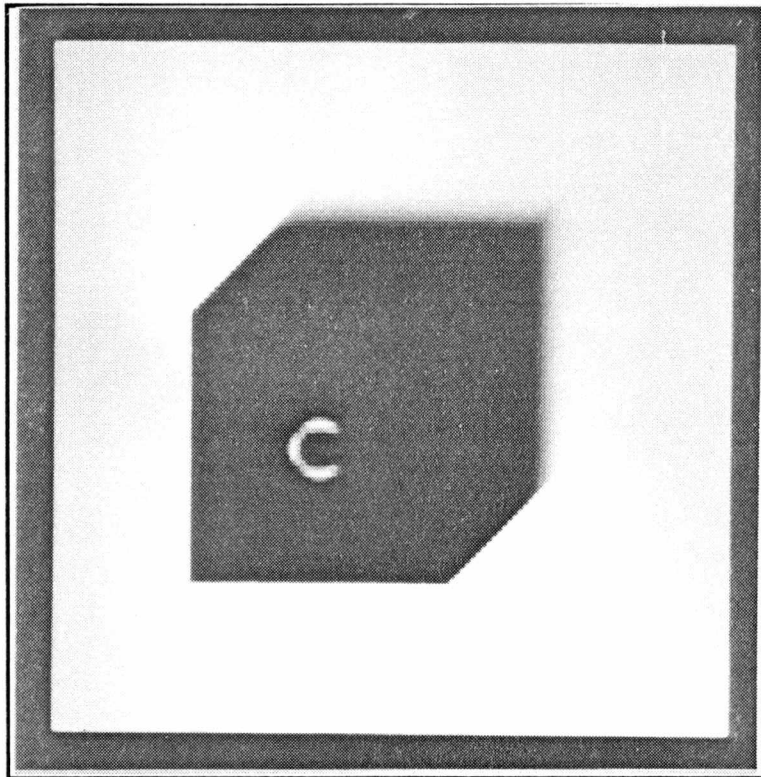


Figure 5.1: Synthetic intensity image.

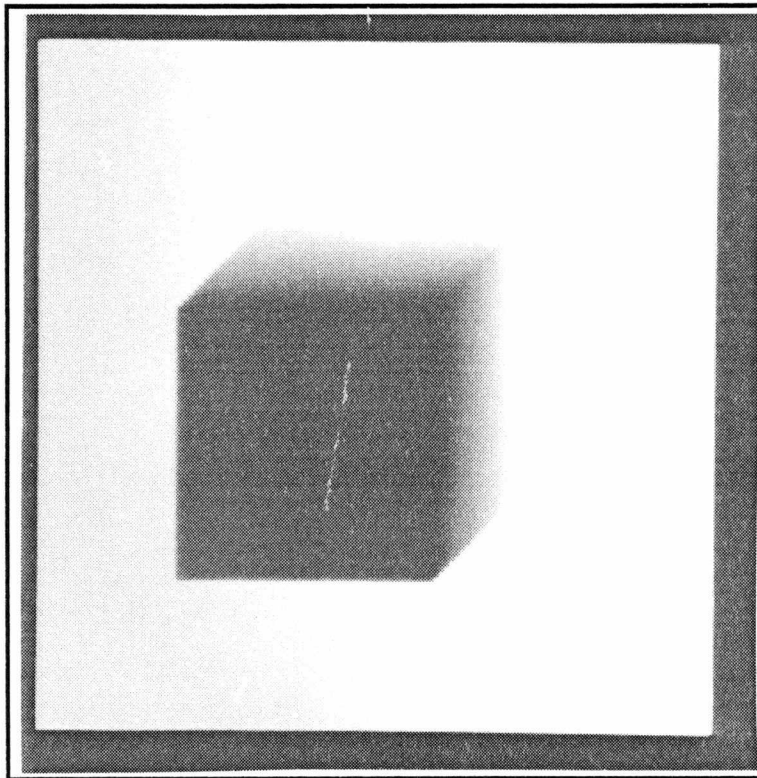


Figure 5.2: Synthetic range image.

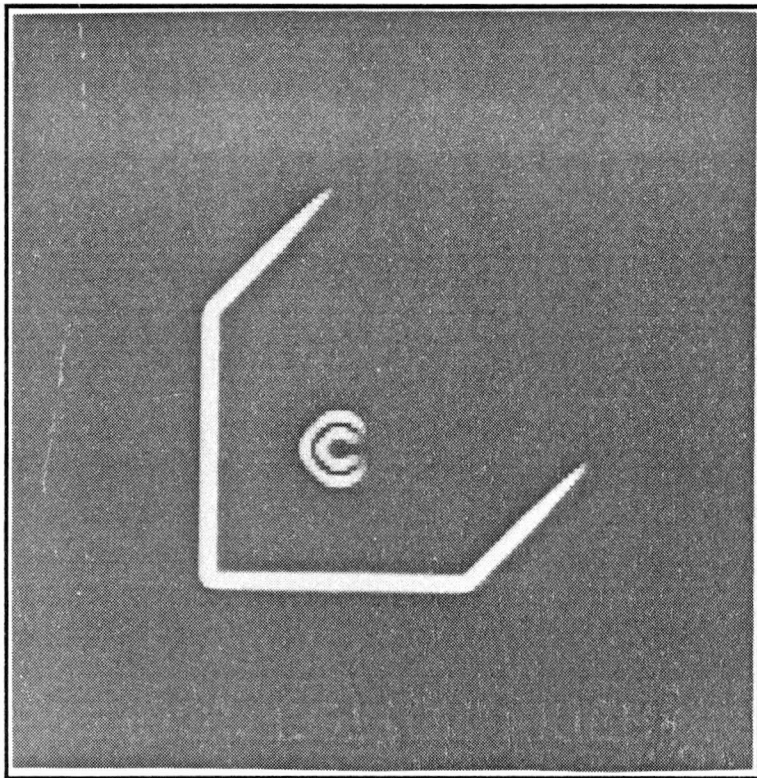


Figure 5.3: Edge map generated from the synthetic intensity image.

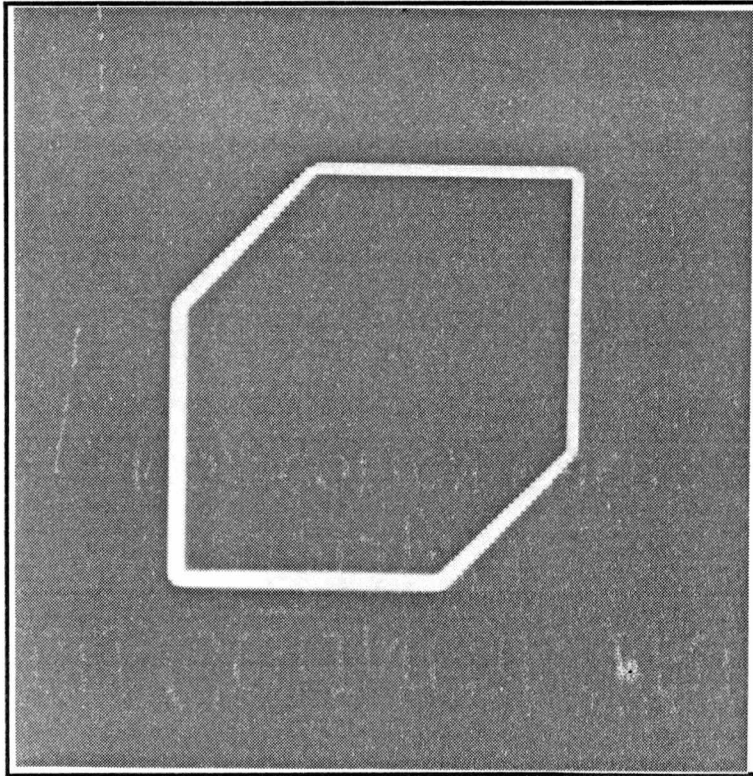


Figure 5.4: Edge map generated from the synthetic range image.

and are hardly visible. The results of the fusion of these two edge maps are shown for the following values of λ^2 : 1, 1.5, 2, 3 and 8. They are displayed in Fig. 5.5 through Fig. 5.9.

5.1.2 Evaluation of the Fusion Scheme

These images allow clear vision of the scope of variation of the algorithm's results. For low values of λ , typically, $\lambda \leq 1$, the method favors the principle of existence. This can be observed in Fig. 5.5. The edges related to the label appear in the fused edge map just as they are in the original intensity edge map. Also, the strong edges appearing in the range data at the intersection of the back side of the cube with the background are selected over the corresponding weaker edges in the intensity data. For these two edges, the fusion function $\alpha(i, j)$ takes either the value zero when selecting the edges from the intensity edge map, or the value one when selecting the range edges. The only area for which the fusion function does not take either of its extreme values is the intersection of the front side of the cube with the background. At this location, both maps feature strong edges, and the fusion function performs an operation close to the arithmetic average. The performance of the algorithm for low values of α could be stated as follows: The algorithm performs an operation similar to an "OR" function. It tends to select the maximum edge value among the two maps.

For large values of λ , typically, $\lambda \geq 4$, the method strongly emphasizes the principle of confirmability. The edges which appear in the output edge map are

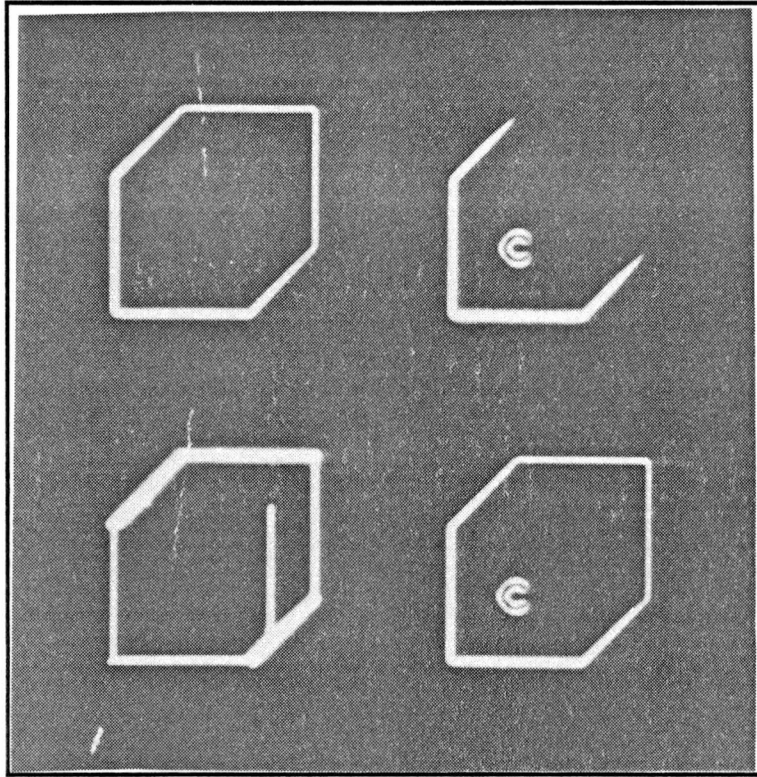


Figure 5.5: Fusion of range and intensity with $\lambda = 1$: original range edge map (top left), original intensity edge map (top right), fusion function (bottom left), and resulting fused edges (bottom right).

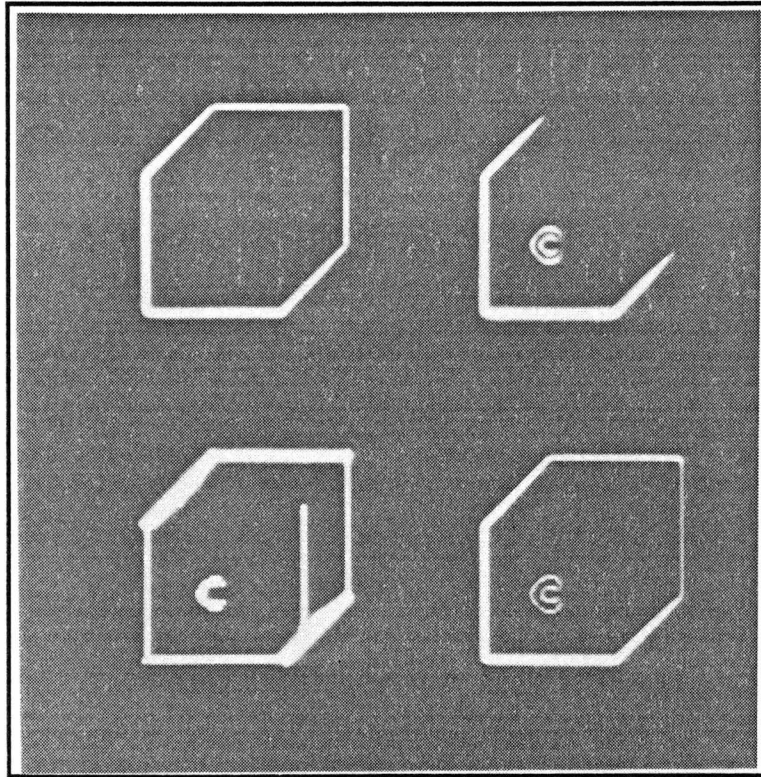


Figure 5.6: Fusion of range and intensity with $\lambda = 1.5$: original range edge map (top left), original intensity edge map (top right), fusion function (bottom left), and resulting fused edges (bottom right).

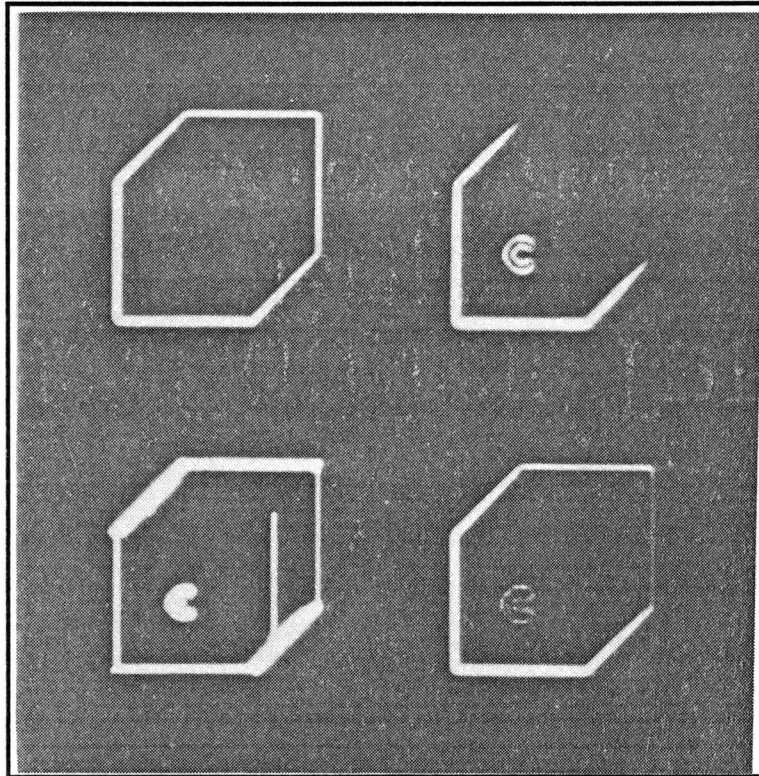


Figure 5.7: Fusion of range and intensity with $\lambda = 2$: original range edge map (top left), original intensity edge map (top right), fusion function (bottom left), and resulting fused edges (bottom right).

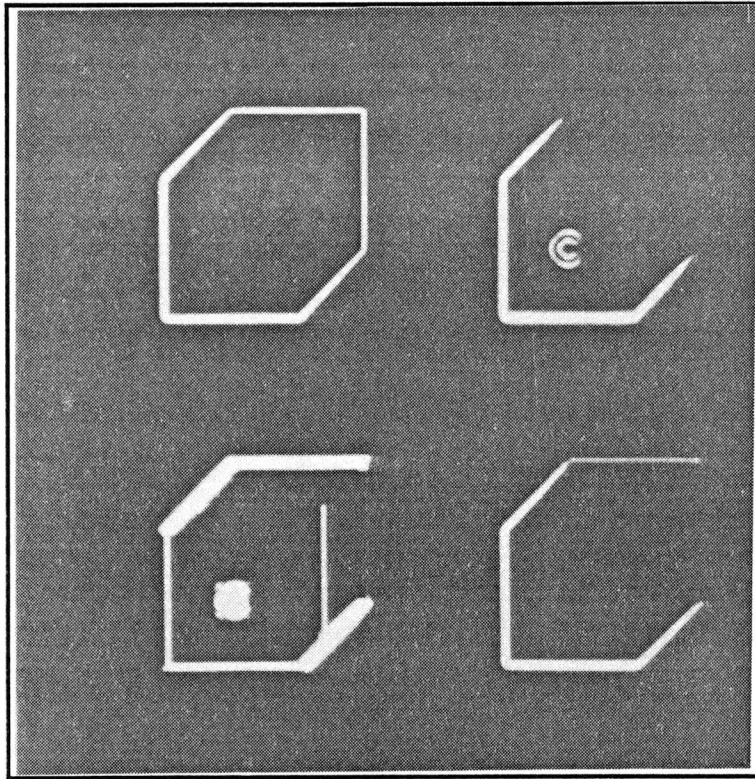


Figure 5.8: Fusion of range and intensity with $\lambda = 3$: original range edge map (top left), original intensity edge map (top right), fusion function (bottom left), and resulting fused edges (bottom right).

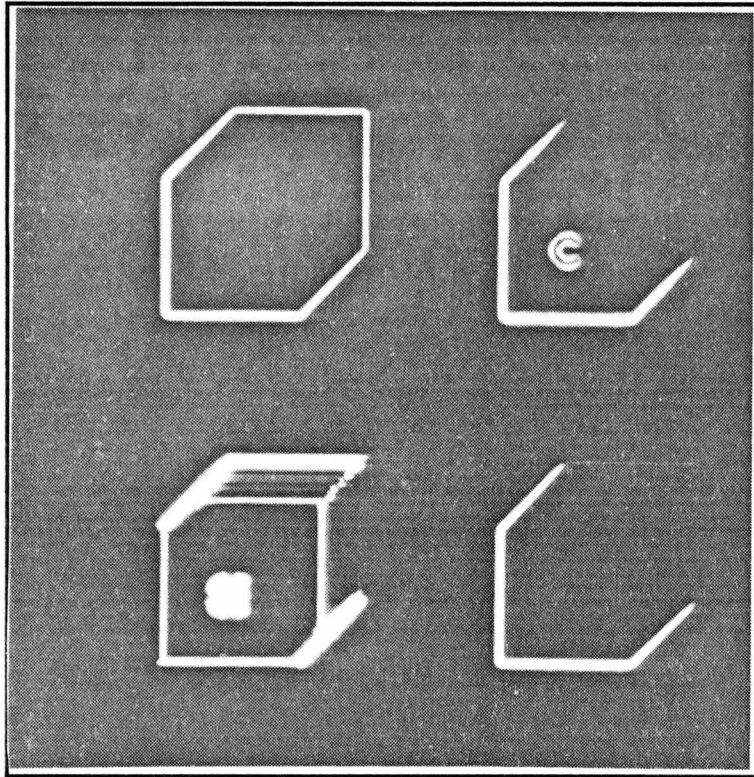


Figure 5.9: Fusion of range and intensity with $\lambda = 8$: original range edge map (top left), original intensity edge map (top right), fusion function (bottom left), and resulting fused edges (bottom right).

those which exist in both images. This result can be seen in Fig. 5.9: The label, which was exclusively present in the intensity edge map, has been eliminated in the fused edge map. The fusion function at that location selects the range data. The interaction between the principles of existence and confirmability is most noticeable for the edges located at the intersection of the back side of the cube with the background. These edges appear fairly degraded if compared with the values they reached in the range edge map, but they also appear significantly stronger than they are in the intensity data. Even though the principle of confirmability is highly stressed, it does not neutralize the effect of the principle of existence. For large values of λ , the fusion function removes the absent edges while still emphasizing low edges if these are confirmed. The edges located at the intersection of the front side of the cube with the background behave much like they did for $\lambda \leq 1$. Obviously, the fusion scheme is not expected to perform in the areas where the inputs do not present any disparity; there is nothing to fuse. Finally, an observation of the fusion function reveals several lines on the lateral sides of the cube. It was mentioned earlier in this chapter that a minimal threshold on the edge intensity must be specified. Most of the gradual changes in the range or intensity data yield edge values which are smaller than this threshold, and therefore, these very weak edges are not fused. These unexpected lines correspond to areas where the edge intensity is just above the threshold. Consequently, the edges are fused, but they are not visible in either of the input maps, nor in the fused edge map. For this reason, these lines should

be considered insignificant. To summarize, with high values of λ , the algorithm produces an output edge map comparable to but also "smarter" than an AND operation.

For values of λ within the range $[1, 4]$, the outputs vary gradually between the extremes. These cases are those for which the α 's are distributed in the range $[0, 1]$, and the resulting fused edge map presents some characteristics which make it different from AND/OR types of operations. As detailed above, the principle of existence is highly predominant for $\lambda = 1$ [Fig 5.5]. The influence of the principle of confirmability begins to be noticeable with $\lambda = 1.5$, as illustrated in Fig. 5.6. By looking at this image, two main tendencies can be observed: (1) The letter "C," which appears exclusively in one of the input edge maps, is substantially reduced in size and intensity in the fused edge map. (2) The intensity of the vertical edge which is located at the intersection of the back side of the cube and the background is decreased. This edge is the range edge which is the least confirmed in the intensity data, and therefore, is the first to be affected by the influence of the principle of confirmability.

These two observations remain valid and are emphasized with increasing values of λ , as shown in Fig. 5.7 and Fig. 5.8. For $\lambda = 2$, only a ghost of the label is present in the fused edge map, and this ghost has almost disappeared for $\lambda = 3$. In addition to these developments, the influence of the principle of confirmability extends to other areas, as for instance the horizontal edge located at the intersection of the back side of the cube with the background.

The power of the algorithm can be illustrated by thresholding the results obtained with $\lambda = 3$. The original edge maps are such that the intensity of the label in the intensity edge map is very similar to the intensity of the edges at the intersection of the back side of the cube and the background in the range data. It is interesting to compare the performance of the algorithm at these two locations. The thresholded version of Fig. 5.8 appears in Fig. 5.10. It is clear in this image that the method allows the disposal of an unconfirmed edge of a given intensity and the retention of a very slightly confirmed edge of the same intensity. This ability is a major strength of the algorithm.

At this point, it would be desirable to test the method on real registered range and intensity images. Unfortunately, this issue cannot be addressed in the frame of this thesis. The reason for this is the need for a performing range edge detector. Range edges in real data cannot be detected by means of a mask. A good range edge detector should perform noise removal and should detect both jump boundaries and edges at the intersection of slanted images.

5.2 Fusion of Edge Maps in Color Images

The algorithm designed to fuse two edge maps can be extended to the simultaneous and the pair-wise fusion of three edge maps. In this section, the pair-wise extension is illustrated. As detailed in the section 4.2, this method comprises two steps. The first consists of fusing two different pairs of inputs and the second

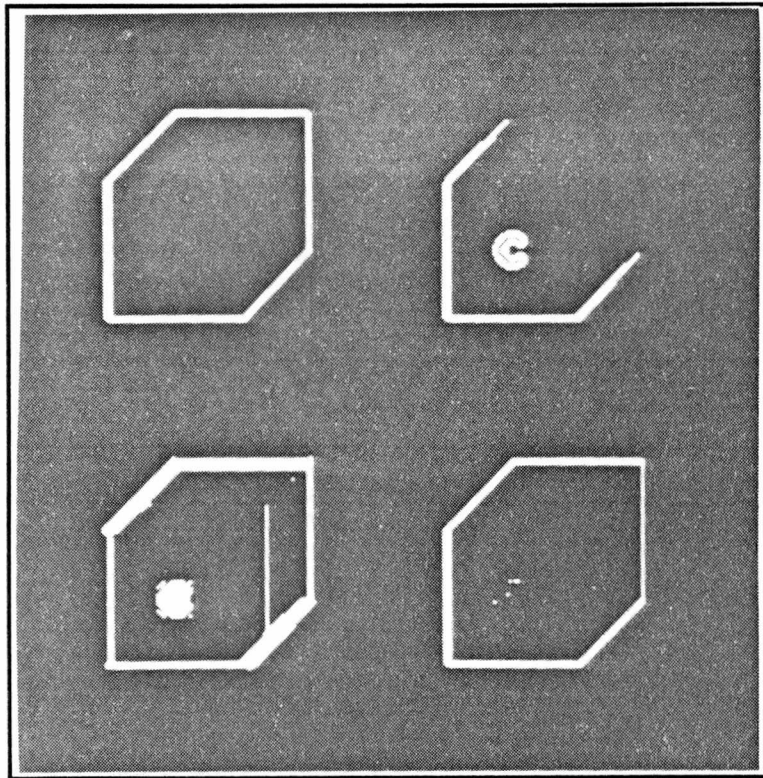


Figure 5.10: Thresholded version of the fusion of range and intensity with $\lambda = 3$.

The threshold equals 45 on the scale 0–255.

of combining their results. The functioning of the algorithm which performs the fusion of two edge maps was illustrated and described in detail in the previous section. Therefore, no synthetic data are utilized in the frame of this extension; only real data are tested. The image which is used throughout this section is represented in two different color bases, and the fusion is performed for these two representations. The use of color information should provide an edge map richer than one which would be derived from brightness data only. It would not be appropriate to select high values of λ . Such a choice would stress the principle of confirmability which, in turn, would emphasize the uncorrelated areas from each input map. The resulting fused edge map would certainly contain less information than the brightness edge map. The principle of existence must be privileged by choosing low values of λ .

5.2.1 Fusion of Color Edge Maps in the RGB Space

The first representation is the widely utilized red, green, and blue (RGB) basis. Fig. 5.11 displays the test picture in this system, and the corresponding scaled edge maps appear in Fig. 5.12.

It can be observed from this image that the strength of the red edge map is concentrated in the branches of the grape vine, in particular those located in the top third of the picture. The blue image features the most defined edges at grape boundaries, but also a very significant overall noise. The green edge map contains edges for both the grape and the vine, while presenting a good overall

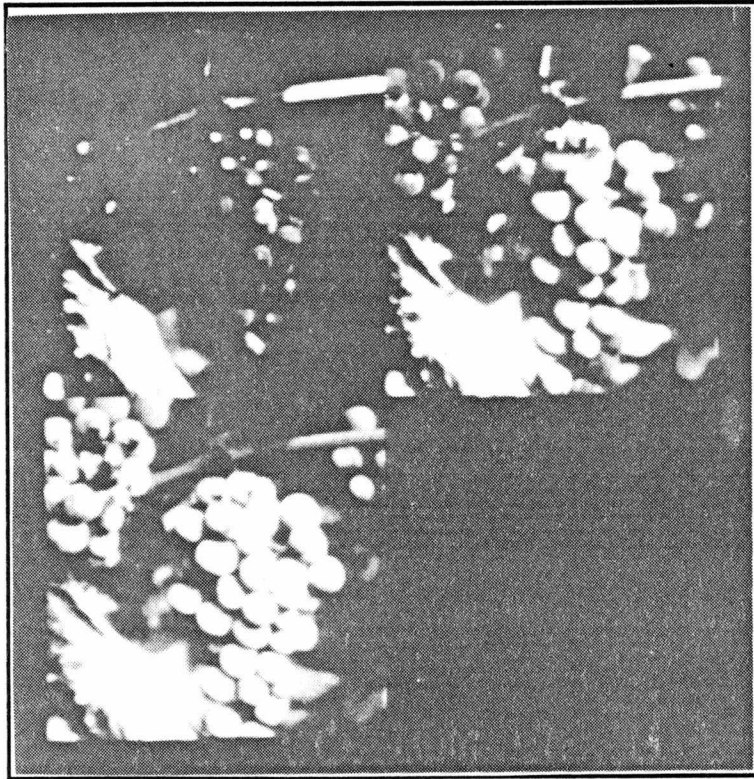


Figure 5.11: Grape image: red component (top left), green component (top right), and blue component (bottom left).

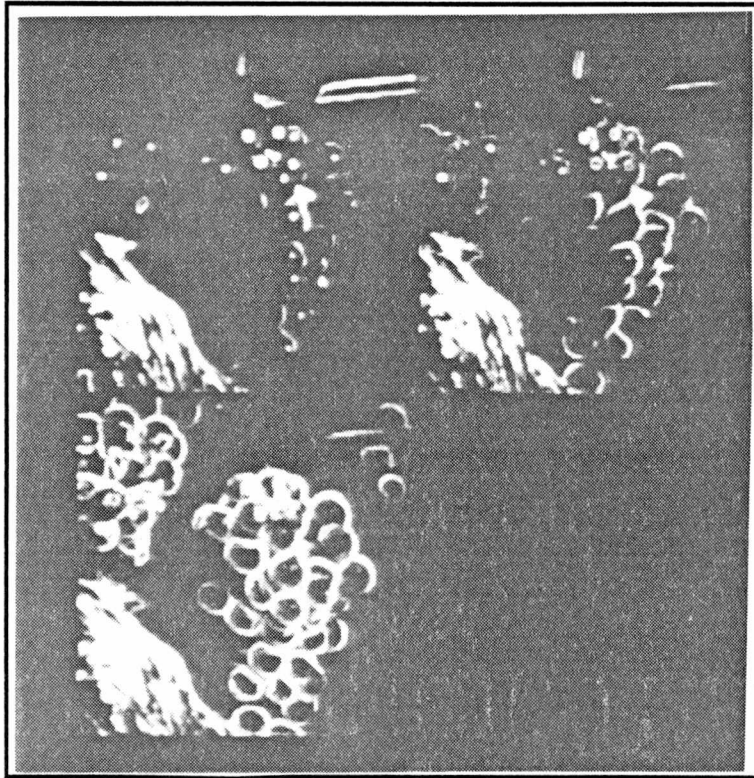


Figure 5.12: Grape image: red edge map (top left), green edge map (top right), and blue edge map (bottom left).

signal to noise ratio, as well. Its strength appears at the leaf boundaries, which are located toward the bottom of the grape bunch.

The RGB basis is known for providing highly correlated data. This property can be easily seen in the edge maps in Fig. 5.12. Even though each picture has a specific edge content, the information they carry appears highly redundant. For instance, the large grape bunch in the middle of the scene has a fairly similar representation in the edge maps derived from the green and the blue images. Due to this very noticeable correlation between the color components and to the fairly high overall edge activity of the test image, the performance of the fusion algorithm may not be as clear as in the case of the synthetic range and intensity images.

The intermediate edge maps appear in Fig. 5.13 and Fig. 5.14. The first one results from the fusion of the red component with the green component, while the second one is derived from the red and blue input edge maps. A low value of λ which favors the principle of existence was selected. These intermediate edge maps are then combined by either calculating their arithmetic or their geometric average. The result which appears in the bottom right corner of Fig. 5.15 is derived by means of the arithmetic average. The red, green, and blue original edge maps are displayed respectively at the top left, top right, and bottom left corner of this same figure. Selecting the geometric average provides a better noise immunity, but also tends to lose edge information, which is not acceptable.

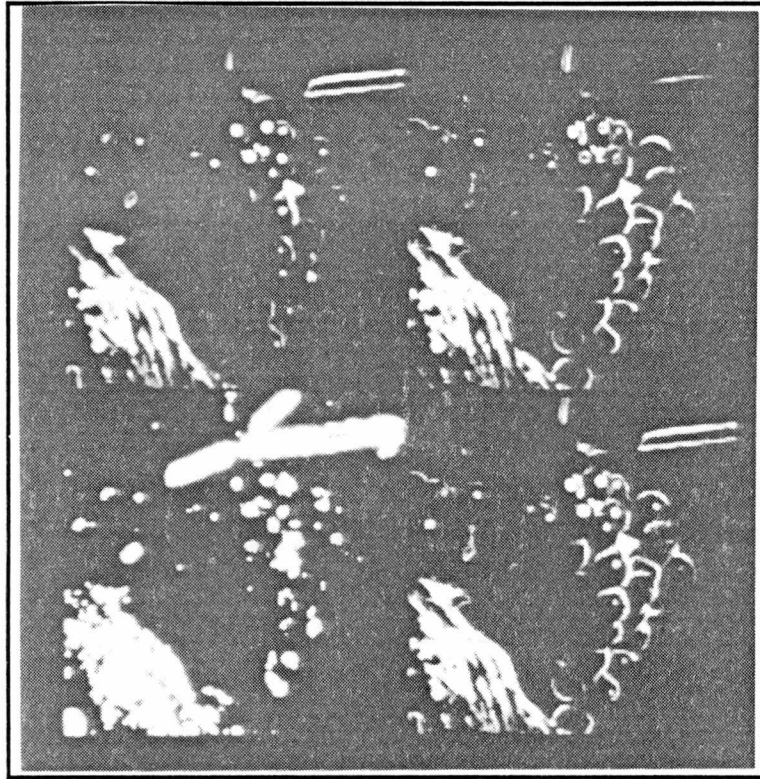


Figure 5.13: Fusion of the red and the green component with $\lambda = .5$: red edge map (top left), green edge map (top right), fusion function (bottom left), and intermediate edge map (bottom right).

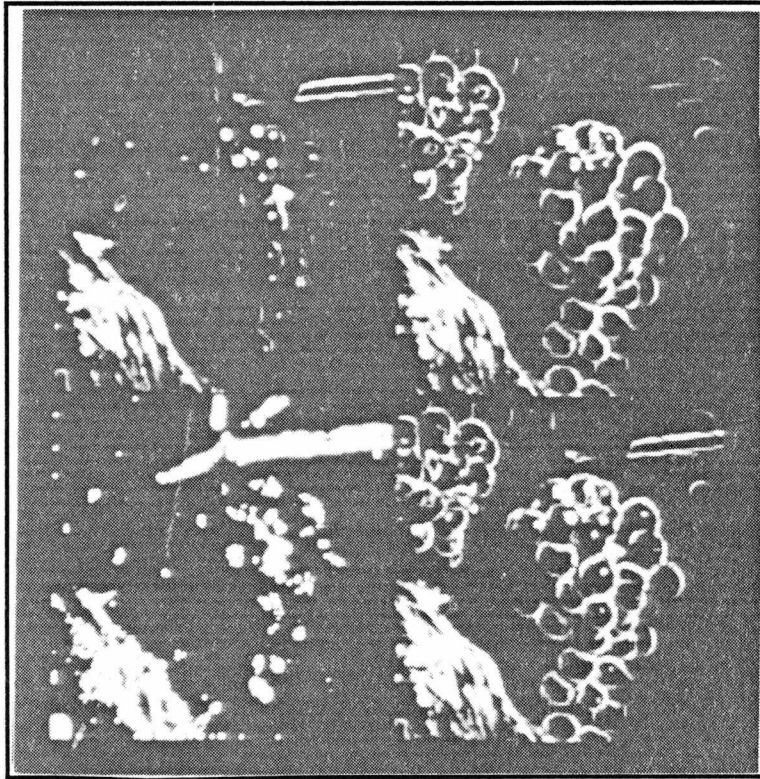


Figure 5.14: Fusion of the red and the blue component with $\lambda = .5$: red edge map (top left), green edge map (top right), fusion function (bottom left), and intermediate edge map (bottom right).

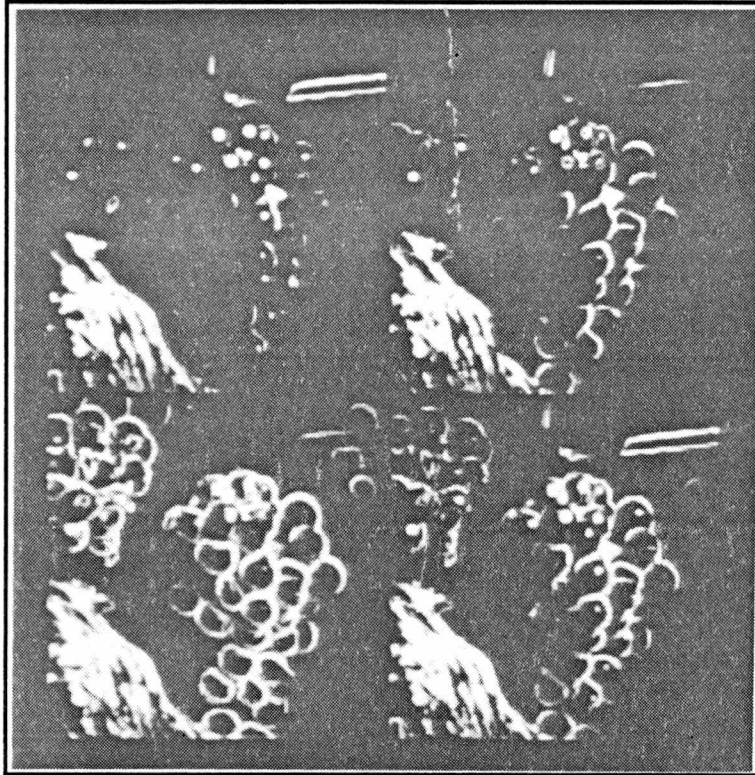


Figure 5.15: Fusion of the red, green, and blue components with $\lambda = .5$: red edge map (top left), green edge map (top right), blue edge map (bottom left), and fused edge map, which is the result of the combination of intermediate edge maps (bottom right).

5.2.2 Fusion of Color Edge Maps in the LT_1T_2 Space

The second base utilized for the representation of color information is the LT_1T_2 , which was introduced by Nevatia [25]. The L component is the luminance component, while T_1 and T_2 contain implicitly the hue and the saturation information. For any pixel location (i, j) ,

$$L(i, j) = .39r(i, j) + .50g(i, j) + .11b(i, j),$$

$$T_1(i, j) = \frac{r(i, j)}{r(i, j) + g(i, j) + b(i, j)},$$

$$T_2(i, j) = \frac{g(i, j)}{r(i, j) + g(i, j) + b(i, j)},$$

where $r(i, j)$, $g(i, j)$, and $b(i, j)$ stand for the red, green, and blue pixel intensity, respectively. Note that the transformation is *not* linear. The grape image is represented in this basis in Fig. 5.16, and the corresponding scaled edge maps are displayed in Fig 5.17. It is clear that the color components are much less correlated than they were in the RGB representation. The luminance component tends to contain all edges, and the T_1 and T_2 components stress several edge areas. For instance, the T_1 component contains mostly edges corresponding to the branches of the grape vine.

The decreased degree of correlation allows for an improved observation of the functioning of the fusion method, as compared to the tests performed in the

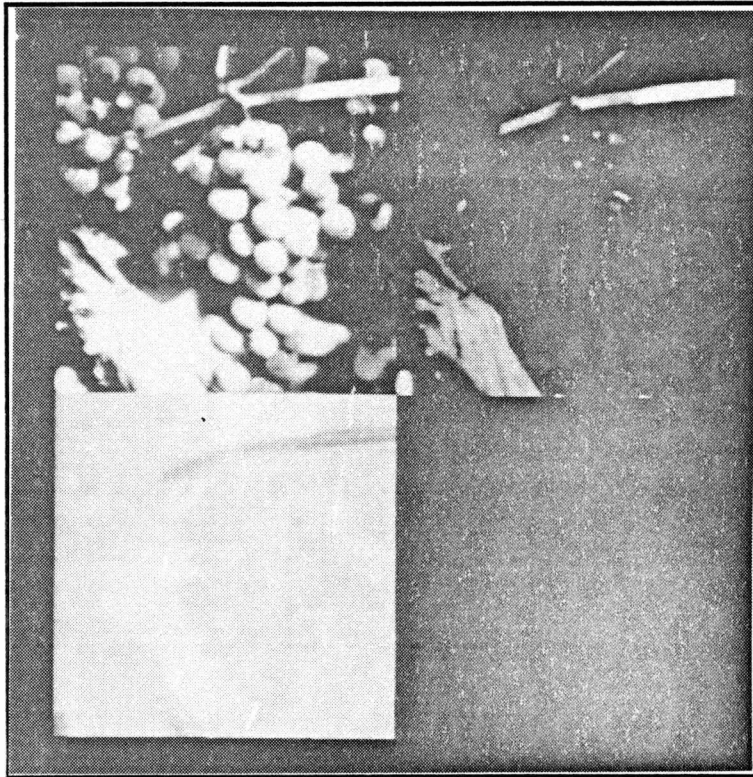


Figure 5.16: Grape image: luminance component (top left), T_1 component (top right), and T_2 component (bottom left).

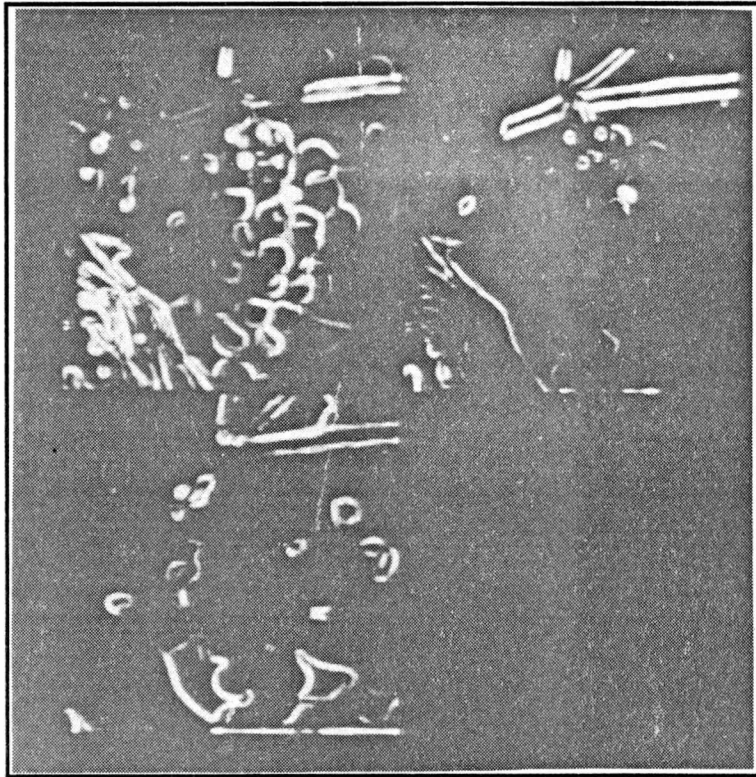


Figure 5.17: Grape image: luminance edge map (top left), T_1 edge map (top right), and T_2 edge map (bottom left).

RGB space. Fig. 5.18 and Fig. 5.19 illustrate the derivation of the intermediate maps. The final edge map, which is obtained by calculating the arithmetic average between the intermediate maps, is displayed in Fig. 5.20.

5.2.3 Evaluation of the Pair-wise Fusion Scheme

As far as the RGB representation is concerned, the observation of Fig. 5.15 shows clearly the visual superiority of the fused edge as compared to any of the edge maps derived from a single color channel. For clarity, the fused edge map which appears in the bottom right corner of Fig. 5.15 is enlarged and displayed in Fig. 5.21. In order to truly appreciate the performance of the algorithm, it is necessary to make some further comparisons. The result will be compared against the geometric average of the inputs $((R + G + B)/3)$ and against the luminance or brightness edge map which would be derived from an original black and white picture. These edge maps are displayed in Figs. 5.22 and 5.23, respectively. Two main achievements of the fusion algorithm can be observed: The edges intensity appears stronger and smoother in Fig. 5.21 than in the other two displayed figures. This gain enhances the edges very significantly. These properties remain valid exclusively for low values of λ (≤ 1). Increasing λ over this value yields a degradation of the final edge map. As mentioned at the beginning of this chapter, increasing the amount of confirmability emphasizes the uncorrelated areas in the input edge maps. This edge degradation can be observed in Fig. 5.24, for which $\lambda = 2$.

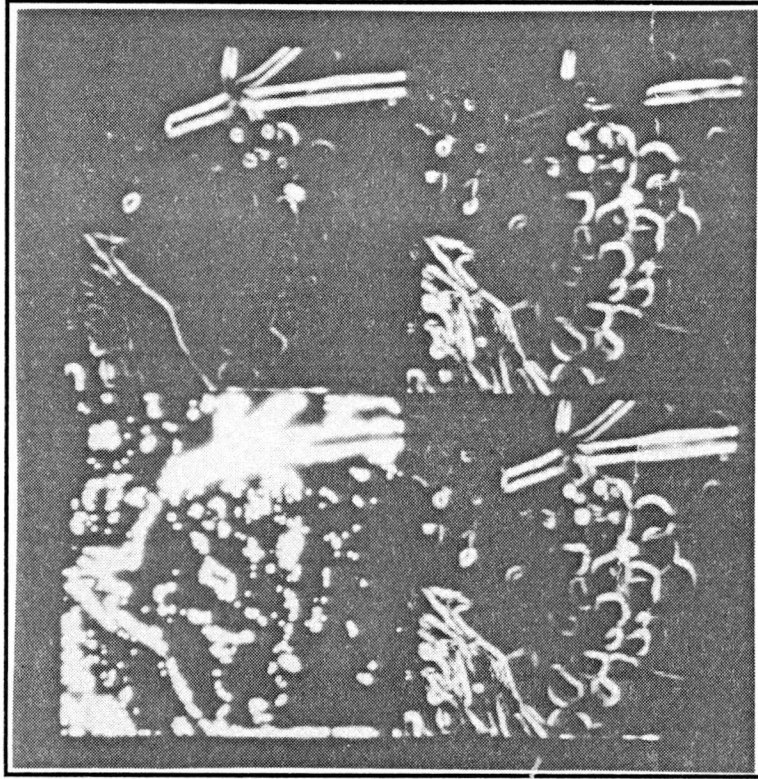


Figure 5.18: Fusion of the T_1 and the luminance components with $\lambda = .5$: T_1 edge map (top left), luminance edge map (top right), fusion function (bottom left), and fused edge map (bottom right).

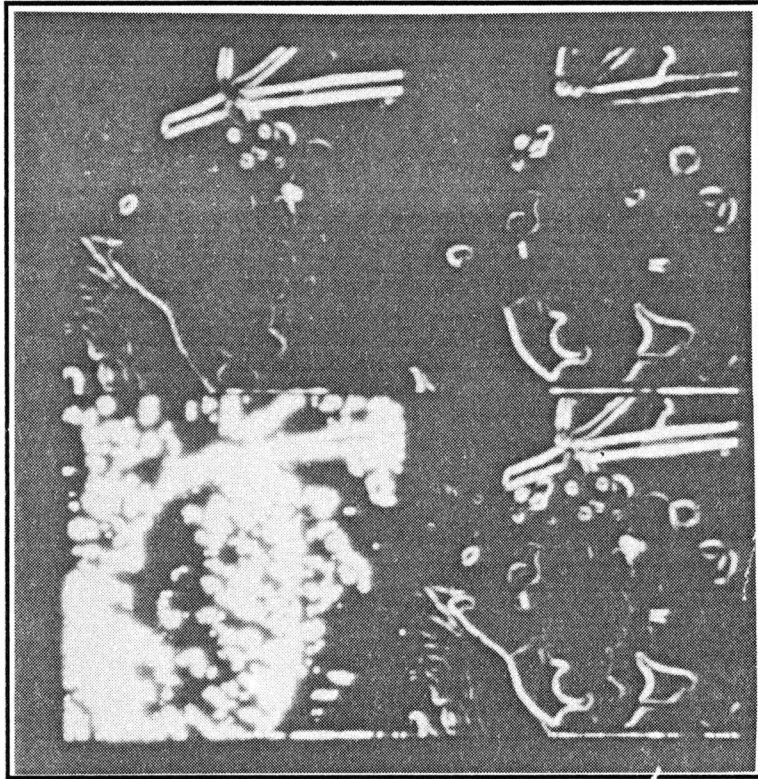


Figure 5.19: Fusion of the T_1 and the T_2 components with $\lambda = .5$: T_1 edge map (top left), T_2 edge map (top right), fusion function (bottom left), and fused edge map (bottom right).

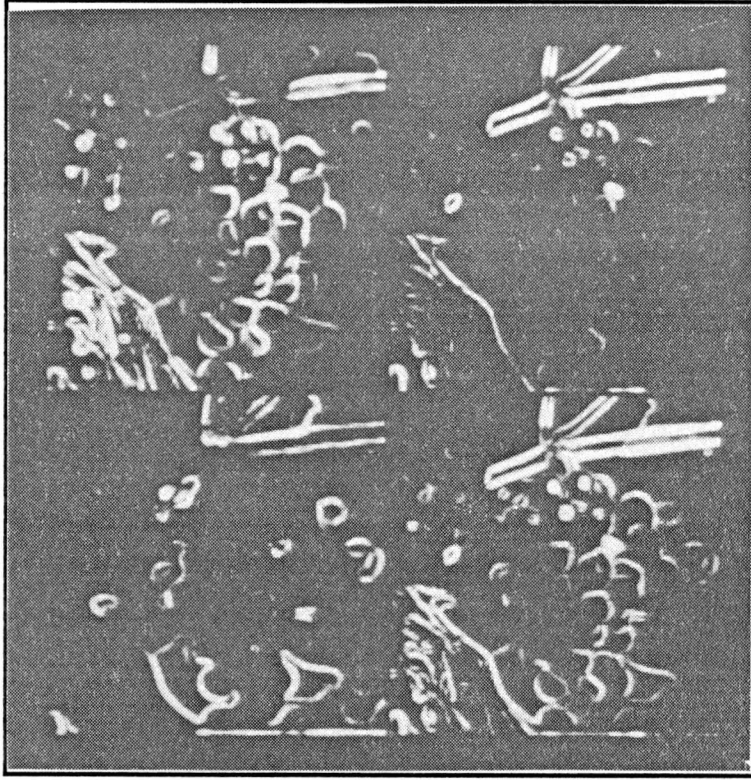


Figure 5.20: Luminance, T_1 , T_2 , and output edge maps with $\lambda = .5$: luminance edge map (top left), T_1 edge map (top right), T_2 edge map (bottom left), and fused edge map, which is the result of the combination of the intermediate edge maps (bottom right).

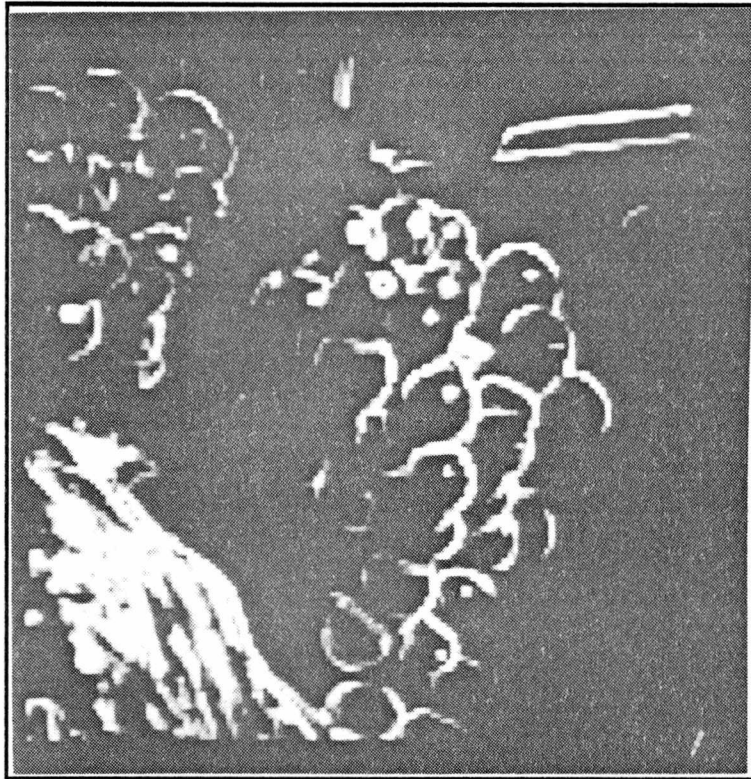


Figure 5.21: Edge map obtained by pair-wise combination of the (red, green) and (red, blue) edge maps with $\lambda = .5$.

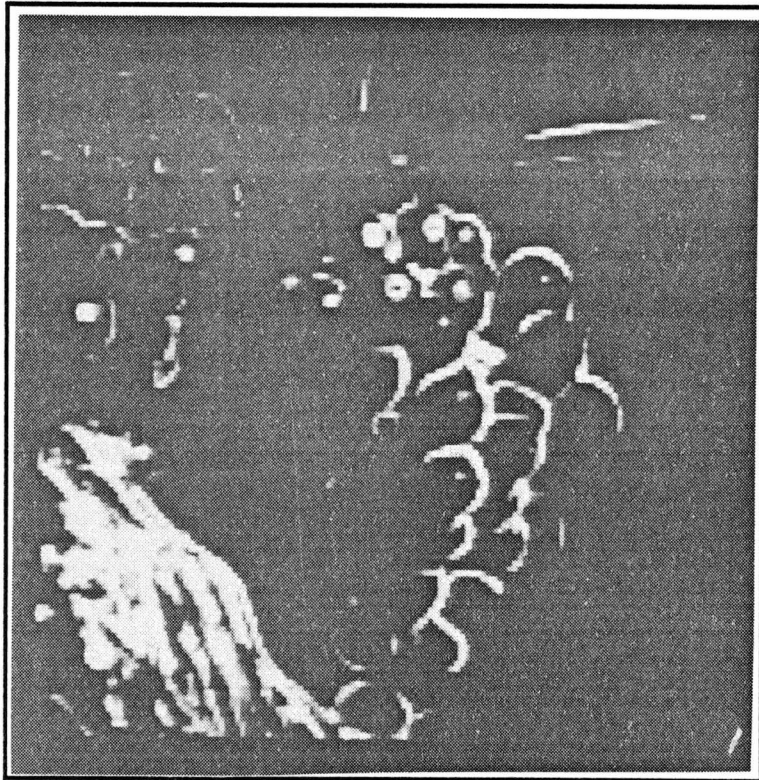


Figure 5.22: Edge map resulting from the arithmetic average of the red, green, and blue edge maps.

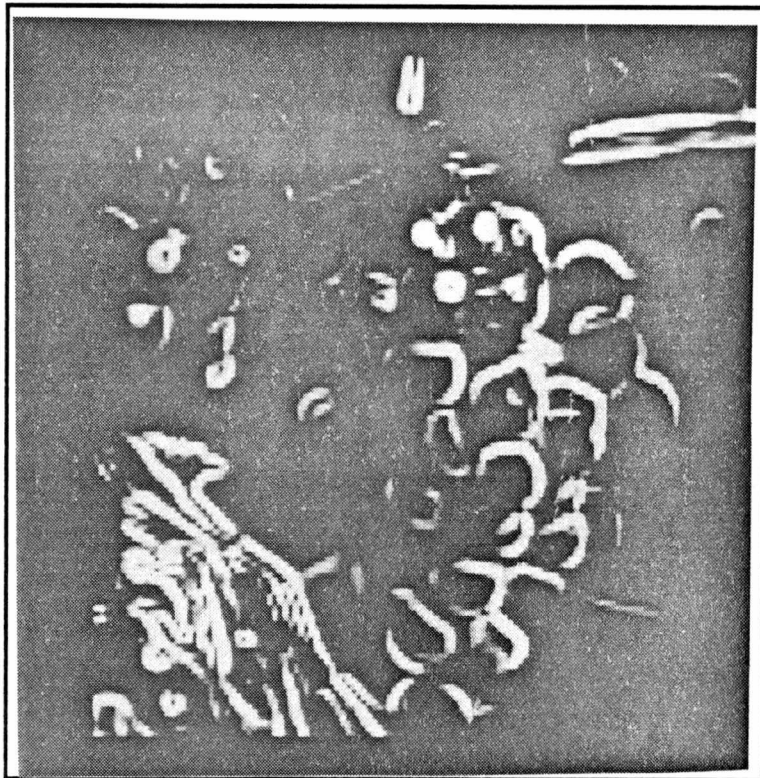


Figure 5.23: Edge map derived from the luminance image.

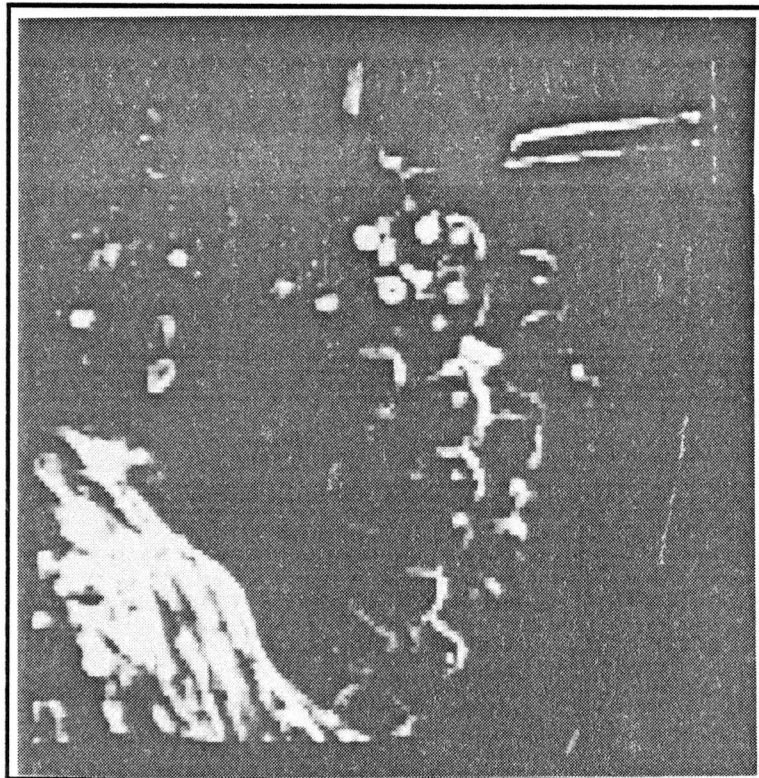


Figure 5.24: Edge map obtained by pair-wise combination of the (red, green) and (red, blue) edge maps with $\lambda = 2$.

For the LT_1T_2 base, the superiority of the fused edge map with regard to the T_1 and the T_2 components can be easily seen in Fig. 5.20. This edge map is enlarged and displayed in Fig. 5.25, allowing a more detailed comparison with the luminance edge map from Fig. 5.23. The fused edge map features stronger edges than the luminance edge map for all the areas where the T_1 or T_2 components present significant edges, but not for the areas where the luminance component is predominant. For these areas, the fusion algorithm induces an attenuation of the edge values. This is noticeable, for instance, in the top left part of the picture, at the level of the small grape. For the purpose of comparison, the arithmetic average of the luminance, the T_1 , and the T_2 components is also displayed [Fig. 5.26]. The edges in the fused edge map appear visually superior to those obtained by calculating the arithmetic average of the three input edge maps.

Other bases have been investigated in the frame of these color experiments, such as the XYZ , LIQ , and $L^*a^*b^*$. The following ideas can be expressed: (1) The fusion algorithm works well when the input edge maps are fairly correlated, as illustrated by means of the RGB space. The fused edge map appears smoother and more complete than the brightness edge map. (2) Applying the algorithm to fairly uncorrelated data induces both gain and loss of information.

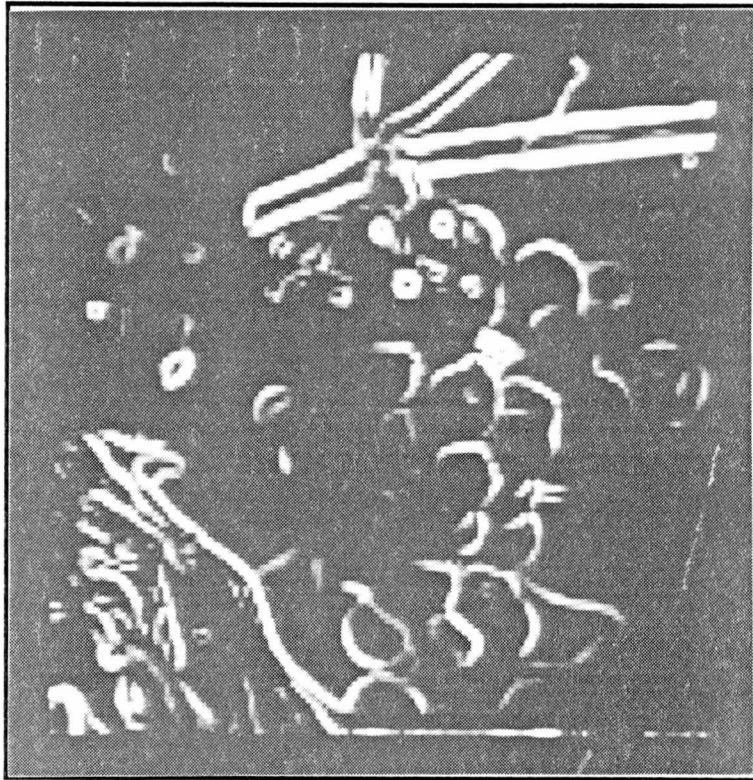


Figure 5.25: Edge map obtained by pair-wise combination of the (T_1, L) and the (T_1, T_2) edge maps with $\lambda = .5$.

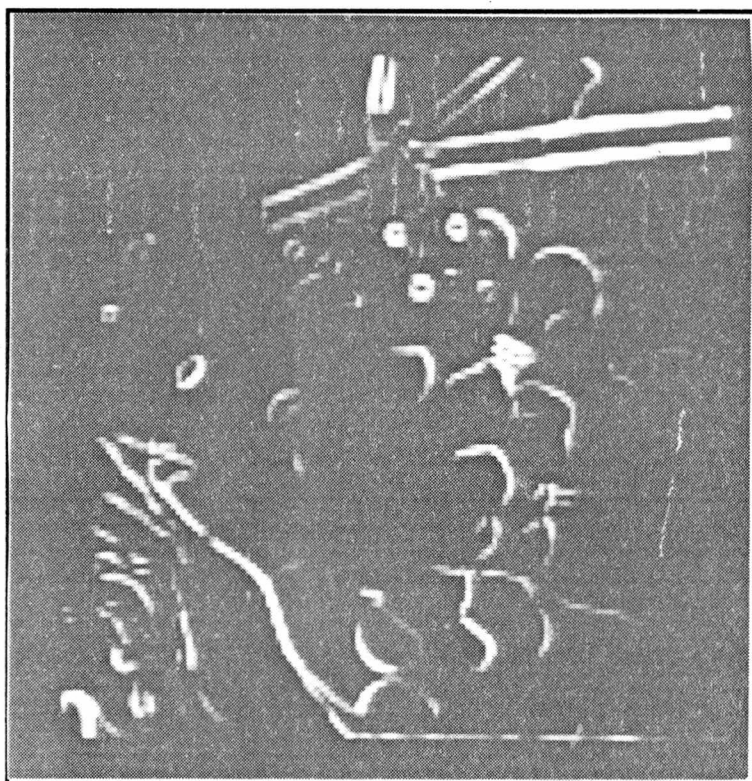


Figure 5.26: Edge map resulting from the geometric average of the luminance, T_1 , and T_2 edge maps.

CHAPTER 6

Conclusions

A new method for low level fusion of registered multisensory edge maps has been presented. The algorithm differs strongly from those previously published in its analytical formulation. The data merging process is expressed by means of a fusion functional which is expanded in Taylor series and truncated. Meaningful mathematically formulated constraints are imposed on the unknown coefficients of the series, and the calculus of variations theory is applied to determine them.

The fusion of multisensory data is illustrated using registered range and intensity edge maps. By varying the weight of the constraints, the fused edge map varies between a result similar to a local OR and a local AND operation between the two edge maps. When the constraints are weighted in a balanced fashion, the algorithm performs a true fusion by selecting contribution from the two inputs simultaneously. This algorithm was extended to the fusion of three edge maps and illustrated by means of color edge maps. The fused edge map contains all edges present in the inputs with an increased smoothness.

Another strength of this algorithm is its generality. It can be extended to other types of registered data. Also, different types of features can be extracted and fused.

BIBLIOGRAPHY

BIBLIOGRAPHY

- [1] H. K. Nishihara and T. Poggio, "Robotics Research," in *The First International Symposium*, (M. Brady and R. Paul, eds.), (Cambridge, MA), pp. 490-505, The MIT Press, 1983.
- [2] C. Brown, J. Aloimonos, M. Swain, P. Chou, and A. Basu, "Texture, Contour, Shape, and Motion," *Pattern Recognition Letters*, vol. 5, pp. 151-168, February 1987.
- [3] B. K. P. Horn, *Robot Vision*. Cambridge, MA: The MIT Press, 1986.
- [4] A. P. Pentland, "A New Sense for Depth of Field," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-9, no. 4, pp. 523-531, July 1987.
- [5] Y. Sato, H. Kitagawa, and H. Fujita, "Shape Measurement of Curved Objects Using Multiple Slit Ray Projections," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-4, no. 6, pp. 641-646, November 1982.
- [6] R. A. Jarvis, "A Perspective on Range Finding Techniques for Computer Vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-5, no. 2, pp. 122-139, March 1983.
- [7] R. Nevatia, *Machine Perception*, ch. Depth Measurement and Analysis, pp. 158-185. 1982.
- [8] P. Besl and R. Jain, "Range Image Understanding," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, (San Francisco, CA), pp. 430-449, June 1985.
- [9] T. D. Garvey, "An Experiment With a System for Locating Objects in Multisensory Images," in *The Third International Joint Conference on Pattern Recognition*, (Coronado, CA), pp. 567-575, November 1976.
- [10] R. O. Duda, D. Nitzan, and P. Barrett, "Use of Range and Reflectance Data to Find Planar Surface Regions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-1, no. 3, pp. 259-271, July 1979.
- [11] R. O. Duda and D. Nitzan, "Low-Level Processing of Registered Intensity and Range Data," in *The Third International Joint Conference on Pattern Recognition*, (Coronado, CA), pp. 598-601, November 1976.

- [12] D. Nitzan, A. E. Brain, and R. O. Duda, "The Measurement and Use of Registered Reflectance and Range Data in Scene Analysis," in *Proceedings of the IEEE on space Data Transmission*, pp. 206-220, February 1977.
- [13] M. J. MaGee, B. A. Boyter, C. Chien, and J. K. Aggarwal, "Experiments in Intensity Guided Range Sensing Recognition of Three-Dimensional Objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-7, no. 6, pp. 629-637, November 1985.
- [14] J. K. Aggarwal and M. J. Magee, "Determining Motion Parameters Using Intensity Guided Range Sensing," *Pattern Recognition*, vol. 19, no. 2, pp. 169-180, April 1986.
- [15] M. J. Magee and J. K. Aggarwal, "Determining Motion Parameters Using Intensity Guided Range Sensing," in *Seventh International Conference on Pattern Recognition*, (Montreal, Canada), pp. 538-541, July/August 1984.
- [16] J. P. Bixler and D. P. Miller, "A Sensory Input System for Autonomous Mobile Robots," in *Proceedings of the 1987 Workshop on Spatial Reasoning and Multi-Sensor Fusion*, (St. Charles, IL), pp. 211-219, October 1987.
- [17] B. Gil, A. Mitiche, and J. K. Aggarwal, "Experiments in Combining Intensity and Range Edge Maps," *Computer Vision, Graphics, and Image Processing*, vol. 21, no. 3, pp. 395-411, March 1983.
- [18] A. Mitiche and J. K. Aggarwal, "Image Segmentation by Conventional and Information-Integrating Techniques: A Synopsis," *Image and Vision Computing*, vol. 3, no. 2, pp. 50-62, May 1985.
- [19] A. Mitiche and J. K. Aggarwal, "Multiple Sensor Integration/Fusion Through Image Processing: A Review," *Optical Engineering*, vol. 25, no. 3, pp. 380-386, March 1986.
- [20] M. J. Magee and J. K. Aggarwal, "Using Multisensory Images to Derive the Structure of Three-Dimensional Objects - A Review," *Computer Vision, Graphics, and Image Processing*, vol. 32, no. 2, pp. 145-157, November 1985.
- [21] R. Gershon, "Aspects of Perception and Computation in Color Vision," *Computer Vision, Graphics, and Image Processing*, vol. 32, no. 2, pp. 244-277, November 1985.
- [22] C. Thorpe, S. Shafer, and T. Kanade, "Vision and Navigation for the Carnegie Mellon Navlab," in *Image Understanding Workshop*, (Los Angeles, CA), pp. 143-152, February 1987.

- [23] V. Torre and T. A. Poggio, "On edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 2, pp. 147–163, March 1986.
- [24] G. S. Robinson, "Color Edge Detection," *Optical Engineering*, vol. 16, no. 5, pp. 479–484, September/October 1977.
- [25] R. Nevatia, "A Color Edge Detector and Its Use in Scene Segmentation," *IEEE Transactions on systems, Man, and Cybernetics*, vol. SMC-7, no. 11, pp. 820–826, November 1977.
- [26] E. Kreyszig, *Advanced Engineering Mathematics*. New York, NY: John Wiley and Sons, Inc, 3 ed., 1972.
- [27] R. Weinstock, *Calculus of Variations*. New York, NY: Dover Publications, Inc., 1974.
- [28] L. E. Elsgolc, *Calculus of Variations*. Reading, MA: Addison-Wesley Publishing Company, Inc., 1962.
- [29] D. K. Faddeev and V. N. Faddeeva, *Computational Methods of Linear Algebra*. San Francisco, CA: W. H. Freeman and Company, 1963.
- [30] I. Gladwell and R. Wait, *A Survey of Numerical Methods for Partial Differential Equations*. Walton Street, Oxford: Clarendon Press, 1979.
- [31] R. C. Gonzalez and P. Wintz, *Digital Image Processing*. Reading, MA: Addison-Wesley Publishing Company, Inc., 2 ed., 1987.

VITA

Christophe Jean-Marie Pierre Delcroix was born in Roubaix, France, on October 17, 1963. He attended Barral High School in Castres and graduated as valedictorian in July, 1981. He entered the Ecole Supérieure d'Ingénieurs en Electrotechnique et Electronique (E. S. I. E. E.) in Paris, France, in the fall of 1981 and obtained the Diplome d'Ingénieurs degree with major in Electrical Engineering and specialization in Telecommunications in June, 1986.

During this course of study, he took part in the Joint European Program in Electrical Engineering through which he completed his fourth year of studies at the University of Karlsruhe, West Germany, and his fifth year at the University of Essex, Colchester, England. He was granted a full scholarship for his studies at these two universities.

He joined the Electrical and Computer Engineering Department at the University of Tennessee as a graduate student in September of 1986. He was a research assistant to Dr. M. A. Abidi from the fall of 1986 to the spring of 1988 and was awarded the Master of Science degree in August, 1988.

During his undergraduate and graduate studies, he was involved in research in the areas of statistical data compression for digital color image storage (University of Essex), motion analysis, and fusion of multisensory data for edge detection (University of Tennessee).