

1-1-1993

# Ultrasonic tissue characterization using texture analysis

P. R. Arul

*Iowa State University*

Follow this and additional works at: <https://lib.dr.iastate.edu/rtd>

 Part of the [Electrical and Computer Engineering Commons](#)

## Recommended Citation

Arul, P. R., "Ultrasonic tissue characterization using texture analysis" (1993). *Retrospective Theses and Dissertations*. 17668.  
<https://lib.dr.iastate.edu/rtd/17668>

This Thesis is brought to you for free and open access by the Iowa State University Capstones, Theses and Dissertations at Iowa State University Digital Repository. It has been accepted for inclusion in Retrospective Theses and Dissertations by an authorized administrator of Iowa State University Digital Repository. For more information, please contact [digirep@iastate.edu](mailto:digirep@iastate.edu).

Ultrasonic tissue characterization using texture analysis

by

P. R. Arul

ISU  
1993  
AT84  
c. 1

A Thesis Submitted to the  
Graduate Faculty in Partial Fulfillment of the  
Requirements for the Degree of  
MASTER OF SCIENCE

Interdepartmental Program: Biomedical Engineering  
Department: Electrical Engineering and Computer Engineering  
Co-majors: Biomedical Engineering  
Electrical Engineering

Signatures have been redacted for privacy

Signatures have been redacted for privacy

Iowa State University  
Ames, Iowa  
1993

Copyright © P. R. Arul, 1993. All rights reserved.

## TABLE OF CONTENTS

<b>CHAPTER 1. INTRODUCTION</b> . . . . .	1
<b>CHAPTER 2. LITERATURE REVIEW</b> . . . . .	3
Ultrasonic Tissue Characterization . . . . .	3
Beef Quality Grading . . . . .	4
Ultrasound for Beef Grading . . . . .	7
Image Texture Analysis . . . . .	9
Spatial Gray-Level Dependence Matrices Method . . . . .	9
SGLDM based Textural Features . . . . .	12
Gray-Level Run Length Matrices Method . . . . .	17
GLRLM based Textural Features . . . . .	18
<b>CHAPTER 3. MATERIALS AND METHODS</b> . . . . .	22
Data Acquisition . . . . .	22
Texture Analysis . . . . .	23
Statistical Analysis . . . . .	24
<b>CHAPTER 4. RESULTS AND DISCUSSION</b> . . . . .	25
Tissue Features . . . . .	25
Correlation Analysis . . . . .	27
SGLDM based Features . . . . .	27

GLRLM based Features . . . . .	27
Regression Models . . . . .	30
<b>CHAPTER 5. CONCLUSION AND RECOMMENDATIONS FOR FUTURE RESEARCH . . . . .</b>	<b>32</b>
<b>BIBLIOGRAPHY . . . . .</b>	<b>34</b>
<b>APPENDIX . . . . .</b>	<b>37</b>
Marbling Score and %Fat Values for Rib Eye Samples . . . . .	37
C Programs for Computing Textural Features . . . . .	38
<b>ACKNOWLEDGEMENTS . . . . .</b>	<b>61</b>

**LIST OF TABLES**

Table 2.1:	Mean ether extractable %fat of the rib eye classified according to their degree of marbling (Savell et al., 1986) . . . . .	8
Table 4.1:	Mutual correlation coefficients between SGLDM based textural features and correlation with marbling score and %fat . .	28
Table 4.2:	Mutual correlation coefficients between GLRLM based textural features and correlation with marbling score and %fat . .	29

## LIST OF FIGURES

Figure 2.1:	Cross section of rib eye between the 12th and 13th ribs. Small flecks of intramuscular fat can be seen in the muscle area. . .	5
Figure 2.2:	Illustrations of the degrees of marbling referred to in the United States standards for grades of carcass beef . . . . .	6
Figure 2.3:	Relationship between marbling, maturity, and carcass quality grade . . . . .	7
Figure 2.4:	B-mode images of the rib eye region in live animals showing differences in speckle scores (Brethour, 1990) . . . . .	10
Figure 2.5:	Resolution cells 1 and 5 are $0^\circ$ nearest neighbors to resolution cell *; resolution cells 2 and 6 are $135^\circ$ nearest neighbors; resolution cells 3 and 7 are $90^\circ$ nearest neighbors; and resolution cells 4 and 8 are $45^\circ$ nearest neighbors to * (Haralick et al., 1973) . . . . .	11
Figure 3.1:	Typical ultrasonic B-mode image of the rib eye muscle in a live animal showing the ROI. a) boundary between hide and subcutaneous fat; b) boundary between subcutaneous fat and rib eye muscle. . . . .	23

<b>Figure 4.1:</b> Correlation between subjectively determined marbling score and chemically determined %fat for the rib eye samples used in this study . . . . .	26
<b>Figure 4.2:</b> Regression models for predicting marbling score and %fat using SGLDM and GLRLM based textural features . . . . .	31
<b>Figure 4.3:</b> Predicted %fat versus actual %fat using SGLDM based textural features . . . . .	31

## CHAPTER 1. INTRODUCTION

Ultrasound, the application of acoustic waves with frequencies from about 20 KHz to several hundred MHz for the study of matter, is widely used in the fields of engineering, agriculture, and medicine. Engineering applications of ultrasound include the non-destructive testing of fabricated structures. Agricultural applications include determination of backfat thickness in beef cattle and swine, pregnancy testing of hogs, and determination of quality of food like the percentage of solid fat in oil, and the ripeness of fruit. In medicine, ultrasound is being used extensively for non-invasively studying soft tissue structures and organs such as the heart, the kidneys, and the liver. The popularity of ultrasound is due to its low cost of operation, ease of use and inherent safety.

All of the above applications make use of the property of ultrasound to be reflected at interfaces between regions of different acoustic properties. The reflected ultrasonic waves, which contain information about the type, size, and structure of the region being studied, can be displayed one-dimensionally (A-mode) or further processed to form an image (B-mode). Frequency shifts in the reflected waves can be used to study objects in motion (Doppler Ultrasound). Good amount of information is obtained by visual inspection and interpretation of the above displays. More recently, advanced signal processing and pattern recognition techniques have been



applied to ultrasonic data to extract parameters that can quantitatively, and hence more accurately, describe and differentiate the regions being studied.

In the United States, commercially available beef is now being graded subjectively by certified United States Department of Agriculture (USDA) inspectors. This is done by visually determining the amount and distribution (marbling) of intramuscular fat in the rib-eye muscle. Since grading is the primary basis for establishing the economic value of beef carcasses, there is a growing demand in the meat industry for a more objective system of evaluating beef carcasses. Also, an ability to estimate marbling in live animals would facilitate clustering procedures for targeting carcasses to more exact packers. In various studies, ultrasound (both A-mode and B-mode) has been shown to have a very good potential for the above tasks.

The objective of this study was to explore the possibility of using texture analysis on ultrasonic images of the rib-eye as a tool for estimating marbling in live beef animals. Two texture analysis methods, namely, the spatial gray level dependence matrices and the gray level run length matrices, were used to extract features. Statistical methods were then used to study the correlation between these texture based features and the intramuscular fat content in the rib-eye muscle.

## CHAPTER 2. LITERATURE REVIEW

### Ultrasonic Tissue Characterization

Tissue properties such as degree of inhomogeneity, scatterer type, size, density, and distribution affect the nature of the reflected ultrasonic signal. Ultrasonic tissue characterization is an objective method which involves the quantitative interpretation of information present in ultrasonic signals for diagnosis and monitoring of tissue. In contrast to tissue characterization, subjective methods qualitatively interpret A- and B-mode features such as rate of change of signal amplitude, image texture, “shadowing” due to attenuating tissue regions, highlighting due to nonattenuating regions, boundary regularity, and echogenicity (Feleppa and Yaremko, 1987). Although such methods are extremely useful, they often are operator and instrument dependent, and do not fully use the information present in the received ultrasonic signals.

Schemes of A-mode signals for tissue characterization include estimation of attenuation, diffraction techniques, and spectrum analysis. Various studies have found attenuation to be a useful tissue-characterization parameter (Amin, 1989; Widyaatmadja, 1991). Also, frequency dependence of attenuation has been used to characterize liver (Narayana and Ophir, 1983), breast tissue (Edmonds et al., 1991), and beef muscle tissue (Chang, 1991). Spectral analysis of the backscattered signal has been used for disease diagnosis and treatment monitoring in ophthalmology, and for

studying diffuse liver diseases (Feleppa and Yaremko, 1987).

Texture analysis has also been used for analyzing B-mode images for tissue characterization. Structures in soft tissue, the dimensions of which are small in comparison with the wavelength of the incident pulse, cause ultrasonic waves to scatter (Rayleigh Scattering). Constructive and destructive interference from phase differences in wave fronts from various scatterers produces a random graininess in ultrasound images that is termed “speckle.” Speckle patterns are affected by scatterer size and density, and the interrogation frequency (Feleppa and Yaremko, 1987). Analysis of speckle texture has been used successfully for differentiating diffuse liver diseases (such as alcoholic liver disease, cirrhosis, and hepatitis) caused by disruption of normal tissue architecture (Lerski et al., 1981; Nicholas et al., 1986; Layer et al., 1990; Wu et al., 1992).

### **Beef Quality Grading**

Commercially available beef is assigned a quality grade by trained graders of the USDA Agricultural Marketing Service—Livestock Division. Quality grades indicate the palatability of beef in terms of tenderness, flavor, and juiciness. Quality grade is determined mainly by two factors—**maturity** and **marbling**. Maturity groups, classified from A to E, under the USDA grading system are determined by subjectively observing the bone structure and color of lean. Age ranges of these maturity groups are as follows:

- A. Approximately 9–30 months
- B. Approximately 30–42 months
- C. Approximately 42–72 months

D. Approximately 72–96 months

E. Over 96 months

Marbling is the content and distribution of intramuscular fat within the lean area of the rib eye between the 12th and 13th ribs (Figure 2.1).



Figure 2.1: Cross section of rib eye between the 12th and 13th ribs. Small flecks of intramuscular fat can be seen in the muscle area.

Degrees of marbling, as shown in Figure 2.2, are subjectively determined and are used to assign quality grade. A marbling score is assigned depending upon the amount of marbling (Figure 2.3). The quality grades for A and B maturity carcasses are USDA Prime, Choice, Select, and Standard. For C and older maturities, the quality grades are Commercial, Utility, and Cutter. The maturity and marbling scores are combined according to the information in Figure 2.3 to determine the final quality grade of a beef carcass. Since maturity is less subjective in nature, the most important factor in quality grading is the marbling.

A more quantitative estimate of marbling levels in the rib eye, called %fat, can be obtained by performing chemical tests. The %fat values, using the ether extraction method, for various degrees of marbling were studied by Savell et al. (1986) and are

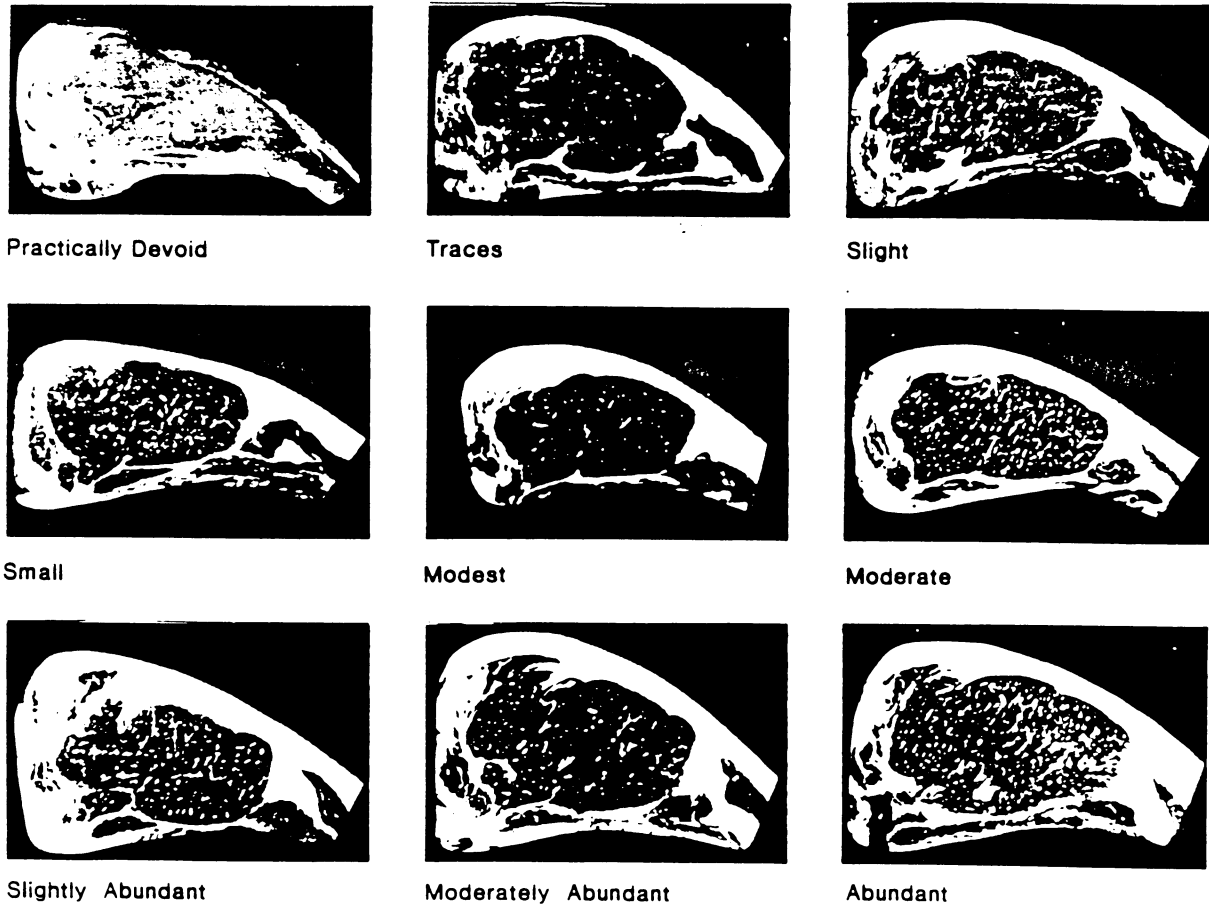


Figure 2.2: Illustrations of the degrees of marbling referred to in the United States standards for grades of carcass beef

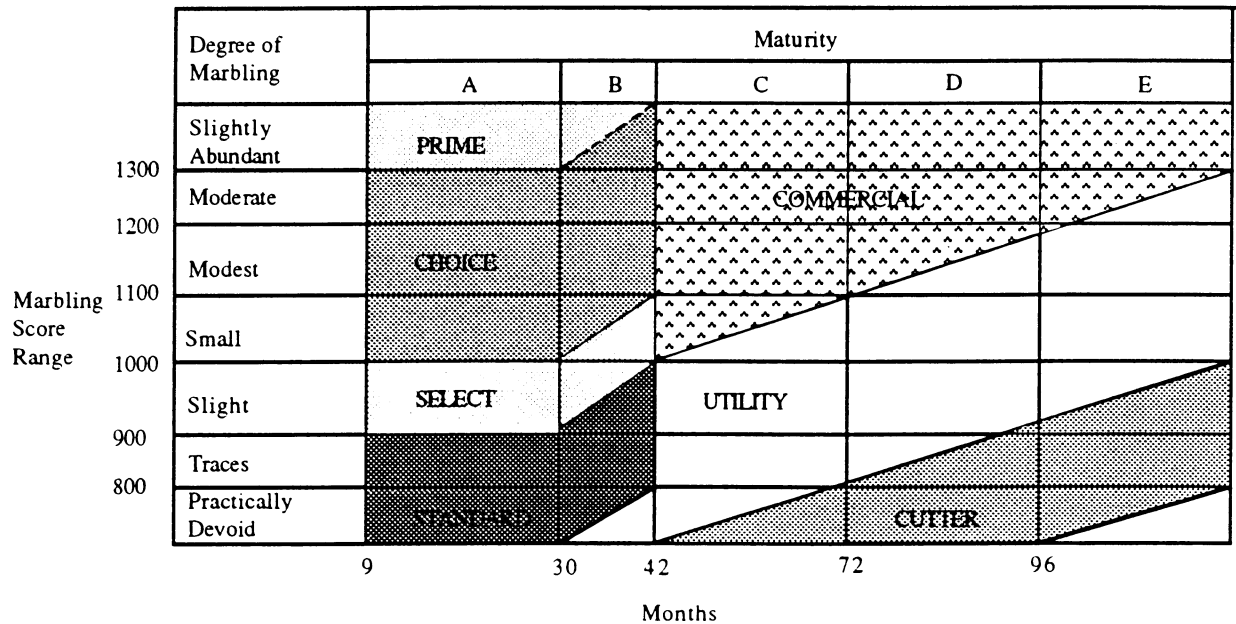


Figure 2.3: Relationship between marbling, maturity, and carcass quality grade

shown in Table 2.1. %Fat values are usually used for correlating with quantitative ultrasonic parameters.

### Ultrasound for Beef Grading

There is a growing demand in the meat industry for an objective quantitative method of quality grading of beef carcasses. Also, a method for estimating marbling in live animals would help cluster feedlot cattle into outcome groups for more effective marketing. Various feasibility studies have found ultrasound to have a good potential for quantitative characterization of marbling in beef carcasses and live animals. An in-vitro ultrasonic Bragg scattering technique has been used for classifying marbling (Haumschild and Carlson, 1983). Estimation of signal attenuation in

Table 2.1: Mean ether extractable %fat of the rib eye classified according to their degree of marbling (Savell et al., 1986)

Degree of Marbling	Ether Extractable %Fat	
	Mean	SD
Prime		
Moderately Abundant	10.42	2.16
Slightly Abundant	8.56	1.60
Choice		
Moderate	7.34	1.50
Modest	5.97	1.15
Small	4.99	1.10
Select		
Slight	3.43	0.89
Standard		
Traces	2.48	0.59
Practically Devoid	1.77	1.12

the rib eye samples using the log-spectral difference method (Amin, 1989) and the slope of the attenuation coefficient (Widyatmadja, 1991) using A-mode signals have been investigated for quantifying marbling. Chang (1991) has reported the use of time domain signatures—power and above-noise threshold count, and a frequency domain signature—the within bandwidth slope, extracted from A-mode signals for in-vitro characterization of marbling. Also, backscattered spectral parameters have shown good potential for evaluating marbling grades (Amin, 1992; Amin et al., 1992). Brethour (1990) reports the development of a scoring system for in-vivo estimation of marbling in live animals based on speckle present in B-mode images of the rib eye muscle.

In this study, texture analysis was used for characterizing marbling in live animals. It has been shown (Brethour, 1990) that as fat deposits (and hence marbling) increase, there is an increase in speckle present in B-mode images (Figure 2.4). Speckle

alters the texture of the image. The ability of texture analysis on B-mode images to differentiate soft tissues has been illustrated by its successful use in studying diffuse diseases of the liver.

### Image Texture Analysis

Texture is an innate property of virtually all surfaces. It contains important information about the structural arrangement of surfaces and their relationship to the surrounding environment. Image texture has long been used for segmentation and classification. A number of approaches for estimating texture have been developed. Two methods that were investigated in this study are 1) the spatial gray-level dependence matrices method (SGLDM), and 2) the gray-level run length matrices method (GLRLM).

#### Spatial Gray-Level Dependence Matrices Method

Haralick et al. (1973) first proposed the SGLDM method and a list of 14 textural features which could be computed from it. This method has been used extensively for identifying diffuse diseases of the liver (Lerski et al., 1981; Nicholas et al., 1986; Layer et al., 1990; and Wu et al., 1992). It is based on the assumption that texture information in an image  $I$  is contained in the overall or "average" spatial relationship which the the gray levels in the image have to one another.

Let an image to be analyzed,  $I$ , have dimensions  $N_y \times N_x$  and  $N_g$  gray levels. Let  $L_x = \{1, 2, \dots, N_x\}$  be the horizontal spatial domain,  $L_y = \{1, 2, \dots, N_y\}$  be the vertical spatial domain, and  $G = \{1, 2, \dots, N_g\}$  be the set of  $N_g$  quantized gray levels. Four angular nearest-neighbor SGLD matrices corresponding to the four principal



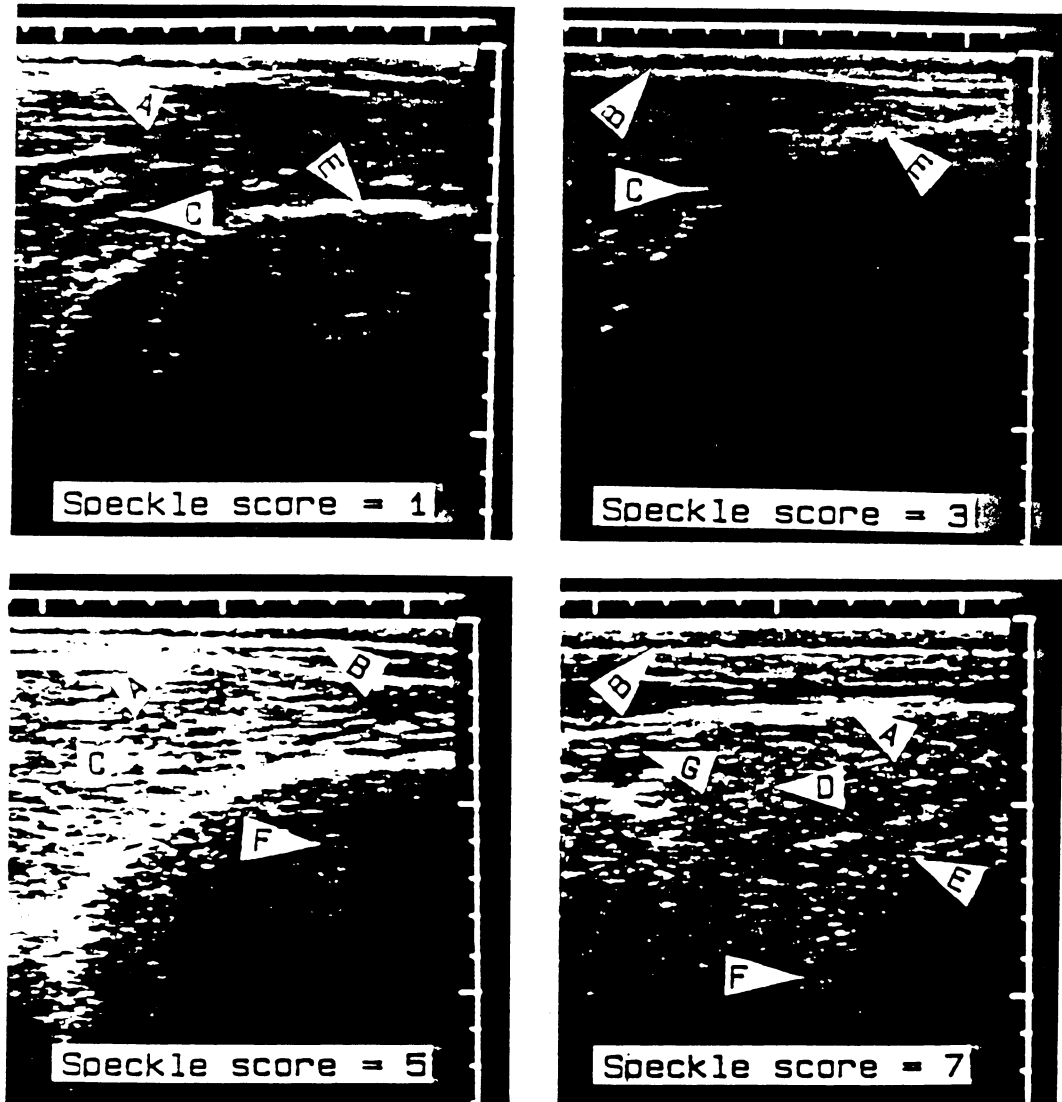


Figure 2.4: B-mode images of the rib eye region in live animals showing differences in speckle scores. A) boundary between subcutaneous fat and rib eye muscle; B) boundary between hide and subcutaneous fat; C) blood vessel; D) area of intense speckle; E) rib bone; F) reverberation from nether region; G) grub (Brethour, 1990)

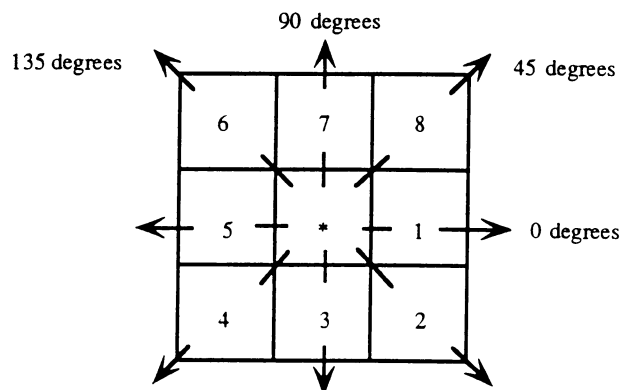


Figure 2.5: Resolution cells 1 and 5 are  $0^\circ$  nearest neighbors to resolution cell \*; resolution cells 2 and 6 are  $135^\circ$  nearest neighbors; resolution cells 3 and 7 are  $90^\circ$  nearest neighbors; and resolution cells 4 and 8 are  $45^\circ$  nearest neighbors to \* (Haralick et al., 1973)

directions (Figure 2.5) are constructed— $P(i, j, d, 0^\circ)$ ,  $P(i, j, d, 45^\circ)$ ,  $P(i, j, d, 90^\circ)$ ,  $P(i, j, d, 135^\circ)$ . A matrix element  $(i, j)$  specifies the number of times two neighboring pixels separated by a distance  $d$ , one with gray level  $i$  and the other with gray level  $j$ , occur on the image in a given direction. The algorithm for computing these matrices is given below:

$$P(i, j, d, 0^\circ) = \#\{((k, l), (m, n)) \in (L_y \times L_x) \\ \times (L_y \times L_x) \mid k - m = 0, |l - n| = d, \\ I(k, l) = i, I(m, n) = j\}$$

$$P(i, j, d, 45^\circ) = \#\{((k, l), (m, n)) \in (L_y \times L_x) \\ \times (L_y \times L_x) \mid (k - m = d, l - n = -d) \\ \text{or } (k - m = -d, l - n = d), \\ I(k, l) = i, I(m, n) = j\}$$

$$\begin{aligned}
P(i, j, d, 90^\circ) &= \#\{((k, l), (m, n)) \in (L_y \times L_x) \\
&\quad \times (L_y \times L_x) \mid |k - m| = d, \\
&\quad l - n = 0, I(k, l) = i, I(m, n) = j\} \\
P(i, j, d, 135^\circ) &= \#\{((k, l), (m, n)) \in (L_y \times L_x) \\
&\quad \times (L_y \times L_x) \mid (k - m = d, l - n = d) \\
&\quad \text{or } (k - m = -d, l - n = -d), \\
&\quad I(k, l) = i, I(m, n) = j\}
\end{aligned}$$

where  $\#$  denotes the number of elements in the set.

Matrices  $P$  are of dimension  $N_g \times N_g$  and are usually normalized to  $p$  according to the number of nearest neighbor pairs. That is

$$p = \left( \frac{P}{R} \right)$$

where

$$R = \begin{cases} 2N_y(N_x - d), & \text{if } \theta = 0^\circ \\ 2N_x(N_y - d), & \text{if } \theta = 90^\circ \\ 2(N_x - d)(N_y - d), & \text{if } \theta = 45^\circ \text{ or } \theta = 135^\circ \end{cases}$$

It is assumed that the textural information is adequately specified by these normalized SGLD matrices.

### SGLDM based Textural Features

Haralick et al. (1973) defined 14 features that can be computed from the SGLD matrices. Mathematical description of these features taken from Haralick et al. (1973)

will be given. The textural information that some of these features represent, from a human perception point of view, is not clearly known. Properties represented by certain features can be intuitively determined. Explanations for the features will be given when known.

**Notation** The following notations will be used to define the textural features:

$p(i, j)$  = the  $(i, j)$ th entry in a normalized SGLD matrix.

$p_x(i)$  = the  $i$ th entry in the marginal probability matrix  
obtained by summing the rows of  $p(i, j)$ .

$N_g$  = the number of gray levels in the image.

$$p_y(j) = \sum_{i=1}^{N_g} p(i, j)$$

$$p_{x+y}(k) = \sum_{\substack{i=1 \\ i+j=k}}^{N_g} \sum_{j=1}^{N_g} p(i, j), \quad k = 2, 3, \dots, 2N_g.$$

$$p_{x-y}(k) = \sum_{\substack{i=1 \\ |i-j|=k}}^{N_g} \sum_{j=1}^{N_g} p(i, j), \quad k = 0, 1, \dots, N_g - 1.$$

$$\mu_x = \sum_{i=1}^{N_g} i p_x(i)$$

$$\mu_y = \sum_{j=1}^{N_g} j p_y(j)$$

$$\sigma_x^2 = \sum_{i=1}^{N_g} (i - \mu_x)^2 p_x(i)$$

$$\sigma_y^2 = \sum_{j=1}^{N_g} (j - \mu_y)^2 p_y(j)$$

**Textural Features** *Angular Second Moment*( $f_1$ ) represents the homogeneity of the image. It is given by the following equation:

$$f_1 = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} [p(i,j)]^2$$

In a homogeneous image, there are very few gray-level changes. Hence, there will be fewer entries of large magnitude in the  $P$  matrices for this image as compared to a large number of entries with small magnitudes for a image with little homogeneity. Therefore, the angular second moment feature for a homogeneous image will have a higher value.

*Contrast*( $f_2$ ) is a measure of the amount of local variations present in an image.

$$f_2 = \sum_{n=0}^{N_g-1} n^2 \left[ \sum_{\substack{i=1 \\ |i-j|=n}}^{N_g} \sum_{j=1}^{N_g} p(i,j) \right]$$

High amounts of local variations in the image give high values for the contrast feature.

*Correlation*( $f_3$ ) is a measure of linear dependencies of intensity values of an image.

$$f_3 = \frac{\sum_{i=1}^{N_g} \sum_{j=1}^{N_g} (ij)p(i,j) - \mu_x \mu_y}{\sigma_x \sigma_y}$$

where  $\mu_x, \mu_y, \sigma_x$ , and  $\sigma_y$  are the means and standard deviations of  $p_x$  and  $p_y$ . Correlation will be much higher for an image with large areas of similar intensities than for an image with noisier, uncorrelated intensities (Shearer and Holmes, 1990).

*Variance* ( $f_4$ ) indicates the variations in image intensity values and is calculated by the following:

$$f_4 = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} (i - \mu_x)^2 p(i, j)$$

For an image with only one gray level, the variance would be zero.

*Inverse Difference Moment* ( $f_5$ ) is calculated by the following:

$$f_5 = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} \frac{1}{1 + (i - j)^2} p(i, j)$$

*Sum Average* ( $f_6$ ) is calculated by the following:

$$f_6 = \sum_{i=2}^{2N_g} i p_{x+y}(i)$$

*Sum Variance* ( $f_7$ ) is calculated by the following:

$$f_7 = \sum_{i=2}^{2N_g} (i - f_6)^2 p_{x+y}(i)$$

*Sum Entropy* ( $f_8$ ) is a measure of the randomness within an image.

$$f_8 = - \sum_{i=2}^{2N_g} p_{x+y}(i) \log(p_{x+y}(i))$$

*Entropy* ( $f_9$ ) is an indication of the complexity of an image and is calculated by the following:

$$f_9 = - \sum_{i=2}^{2N_g} p(i, j) \log(p(i, j))$$

*Difference Variance* ( $f_{10}$ ) is calculated by the following:

$$f_{10} = \text{variance of } p_{x-y}$$

*Difference Entropy* ( $f_{11}$ ), like sum entropy, is also a measure of the randomness in an image.

$$f_{11} = - \sum_{i=0}^{N_g-1} p_{x-y}(i) \log(p_{x-y}(i))$$

*Information Measures of Correlation* ( $f_{12}, f_{13}$ ) are calculated by the following:

$$f_{12} = \frac{HXY - HXY1}{\max[HX, HY]}$$

and

$$f_{13} = (1 - \exp[-2.0(HXY2 - HXY)])^{\frac{1}{2}}$$

where

$$HXY = - \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i, j) \log(p(i, j)),$$

HX and HY are entropies of  $p_x$  and  $p_y$ , and

$$HXY1 = - \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i, j) \log(p_x(i)p_y(j))$$

$$HXY2 = - \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p_x(i)p_y(j) \log(p_x(i)p_y(j))$$

*Maximum Correlation Coefficient* ( $f_{14}$ ) is calculated by the following:

$$f_{14} = (\text{Second largest eigenvalue of } Q)^{\frac{1}{2}}$$

Where

$$Q(i, j) = \sum_{k=0}^{N_g} \frac{p(i, k)p(j, k)}{p_x(i)p_y(k)}$$

## Gray-Level Run Length Matrices Method

The GLRLM method was proposed by Galloway (1975) as a texture analysis tool. Five textural features were defined which could be computed from the run length matrices and were used in the classification of images from nine terrain types. These features are: short run emphasis, long run emphasis, gray level distribution, run length distribution, and run percentage. Chu et al. (1990) proposed two additional features based on run length matrices to complement the earlier defined feature list. They are: low gray-level run emphasis and high gray-level run emphasis. Further, Dasarathy and Holder (1991) proposed four more textural features which were used successfully in the classification of cell images (Dasarathy, 1992). These features are: short run low gray-level emphasis, short run high gray-level emphasis, long run high gray-level emphasis, and long run low gray-level emphasis. So, the set of 11 features was used in this study.

A gray level run is defined as a set of consecutive, collinear points in the image having the same gray level value (Galloway, 1975). The length of the run is the number of image points in the run. For a given image, four GLRL matrices can be computed corresponding to the four principal directions (Figure 2.5)— $p(i, j, 0^\circ)$ ,  $p(i, j, 45^\circ)$ ,  $p(i, j, 90^\circ)$ ,  $p(i, j, 135^\circ)$ . The matrix element  $(i, j)$  specifies the number of times that the image contains a run of length  $j$ , in the given direction, consisting of points having gray level  $i$  (Galloway, 1975). The example below, taken from Galloway (1975), shows a  $4 \times 4$  image having four gray levels (0-3) and the resulting GLRL matrices for the four principal directions.



0 1 2 3  
 0 2 3 3  
*Image :*  
 2 1 1 1  
 3 0 3 0

		<i>Runlength</i>						<i>Runlength</i>					
		0°	1	2	3	4			45°	1	2	3	4
<i>Graylevel</i>	0	4	0	0	0	0			0	4	0	0	0
	1	1	0	1	0	0			1	4	0	0	0
	2	3	0	0	0	0			2	0	0	1	0
	3	3	1	0	0	0			3	3	1	0	0
		90°	1	2	3	4			135°	1	2	3	4
<i>Graylevel</i>	0	2	1	0	0	0			0	4	0	0	0
	1	4	0	0	0	0			1	4	0	0	0
	2	3	0	0	0	0			2	3	0	0	0
	3	3	1	0	0	0			3	5	0	0	0

The number of calculations for computing these matrices is directly proportional to the size of the image.

### GLRLM based Textural Features

The mathematical representation for the 11 textural features that can be computed from the GLRL matrices will be given below. Let

$N_g$  be the number of gray levels in the image,

$N_r$  be the number of different run lengths that occur (so that the dimension of the  $p$  matrix is  $N_g \times N_r$ ), and

$P$  be the number of pixels in the image.

$n_r$ , the total number of runs in the image, is used as a normalizing factor and is calculated by the following:

$$n_r = \sum_{i=1}^{N_g} \sum_{j=1}^{N_r} p(i, j)$$

**Textural Features** *Short Run Emphasis* ( $r_1$ ) emphasizes short runs in the image because of the presence of length of the run squared ( $j^2$ ) in the denominator.

$$r_1 = \frac{1}{n_r} \sum_{i=1}^{N_g} \sum_{j=1}^{N_r} \frac{p(i, j)}{j^2}$$

*Long Run Emphasis* ( $r_2$ )

$$r_2 = \frac{1}{n_r} \sum_{i=1}^{N_g} \sum_{j=1}^{N_r} j^2 p(i, j)$$

Because of the presence of length of the run squared ( $j^2$ ) in the numerator, this feature emphasizes long runs.

*Gray Level Distribution* ( $r_3$ ) measures the gray level nonuniformity of the image.

$$r_3 = \frac{1}{n_r} \sum_{i=1}^{N_g} \left( \sum_{j=1}^{N_r} p(i, j) \right)^2$$

When runs are equally distributed throughout the gray levels, the feature takes on its lowest values. High run length values contribute most to the function (Galloway, 1975).

*Run Length Distribution* ( $r_4$ ) is a measure of the nonuniformity of the run lengths.

$$r_4 = \frac{1}{n_r} \sum_{j=1}^{N_r} \left( \sum_{i=1}^{N_g} p(i, j) \right)^2$$

This feature will take low values when runs are equally distributed throughout the lengths.

*Run Percentage* ( $r_5$ ) is the ratio of the total number of runs to the total number of possible runs if all runs had a length of one.

$$r_5 = \frac{n_r}{P}$$

$r_5$  will take low values for images with good linear structure.

Chu et al. (1990) proposed the next two features to more explicitly characterize the gray level distributions in the image.

*Low Gray-level Run Emphasis* ( $r_6$ )

$$r_6 = \frac{1}{n_r} \sum_{i=1}^{N_g} \sum_{j=1}^{N_r} \frac{p(i, j)}{i^2}$$

*High Gray-level Run Emphasis* ( $r_7$ )

$$r_7 = \frac{1}{n_r} \sum_{i=1}^{N_g} \sum_{j=1}^{N_r} i^2 p(i, j)$$

Dasarathy and Holder (1991) proposed the next four features to emphasize the joint distribution properties of the run lengths and gray levels instead of individual ones separately.

*Short Run Low Gray-level Emphasis* ( $r_8$ )

$$r_8 = \frac{1}{n_r} \sum_{i=1}^{N_g} \sum_{j=1}^{N_r} \frac{p(i, j)}{i^2 j^2}$$

*Short Run High Gray-level Emphasis ( $r_9$ )*

$$r_9 = \frac{1}{n_r} \sum_{i=1}^{N_g} \sum_{j=1}^{N_r} \frac{i^2 p(i, j)}{j^2}$$

*Long Run High Gray-level Emphasis ( $r_{10}$ )*

$$r_{10} = \frac{1}{n_r} \sum_{i=1}^{N_g} \sum_{j=1}^{N_r} i^2 j^2 p(i, j)$$

*Long Run Low Gray-level Emphasis ( $r_{11}$ )*

$$r_{11} = \frac{1}{n_r} \sum_{i=1}^{N_g} \sum_{j=1}^{N_r} \frac{j^2 p(i, j)}{i^2}$$

## CHAPTER 3. MATERIALS AND METHODS

### Data Acquisition

Ultrasonic B-mode images from several hundred live beef animals were acquired over a two year period by staff at the Animal Science Department and Animal Science Extension of Iowa State University. This study includes results from the analysis of the first set of 23 images representing 23 different animals. A portable medical diagnostic ultrasound imaging unit (model Aloka 500V<sup>1</sup>) with a 3.5 MHz 17 cm linear array transducer was used for field scanning. Images of the rib eye muscle between the 12th and 13th ribs were first acquired and stored on a video tape and later were digitized in the lab using a personal computer equipped with a frame grabber board. The digitized images were of size 410×246 pixels and had 256 gray levels. After slaughter, the carcasses were evaluated by a qualified USDA grader to determine the marbling scores. Also, % fat in the rib eye samples were chemically determined by the n-hexane extraction method.

---

<sup>1</sup>Corometrics Medical Systems, Inc., Wallingford, CT.

### Texture Analysis

All the images were transferred to a DEC<sub>v</sub>-5000<sup>2</sup> workstation for further processing. PV-WAVE Point& Click<sup>3</sup>, a commercial signal\image processing software package, was used for displaying the images and for converting them to ASCII format. Since each image contained hide, subcutaneous fat, rib bone, and other anatomical parts apart from the rib eye muscle, macros were constructed in Point&Click to select a region of interest(ROI). The macros enabled the automatic selection of the same ROI in all images. The time required to complete texture analysis on a image depends on the size of the ROI. A ROI of size 80×80 (Figure 3.1) was selected subjectively so that it contained good texture information and did not require much computation time.

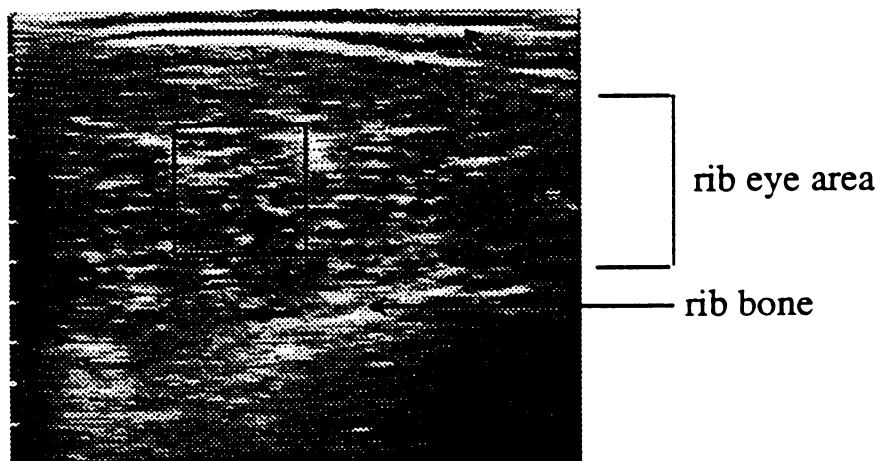


Figure 3.1: Typical ultrasonic B-mode image of the rib eye muscle in a live animal showing the ROI. a) boundary between hide and subcutaneous fat; b) boundary between subcutaneous fat and rib eye muscle.

---

<sup>2</sup>Digital Equipment Corporation, Maynard, MA.

<sup>3</sup>Precision Visuals Inc., Boulder, CO.

Twenty four textural features were calculated (13 based on the SGLDM method with  $d = 1$  and 11 based on the GLRLM method) for the ROI using software written in the C programming language (program listing is given in the Appendix). The SGLDM based feature, maximum correlation coefficient, was not used in the study because the method for computing the eigen values would sometimes fail to converge.

### Statistical Analysis

Statistical analysis of the texture features was done using the SAS<sup>4</sup> package. Correlation analysis was done to find the mutual correlation coefficients ( $\rho$ ) among the textural features. Features with high mutual correlations essentially provide similar information about tissue properties. Features that significantly correlated with %fat or marbling score ( $p < 0.05$ ) were used for further analysis.

Multiple regression methods were used, for systematic analysis of textural features, to find the best regression model for predicting %fat and marbling score. The criteria considered for selecting the optimum model were maximum coefficient of determination ( $R^2$ ), minimum root mean square error (for adequacy of the model) and a  $p$  value less than 0.05 (for significance of the model).

---

<sup>4</sup>SAS Institute, Inc., Cary, NC.

## CHAPTER 4. RESULTS AND DISCUSSION

First, the correlation between marbling scores and %fat values for the 23 samples will be presented. Then the results of the correlation analysis done on the 24 textural features will be discussed. Finally the optimum prediction models that were derived using multiple regression methods will be presented.

### Tissue Features

Marbling scores and %fat values for all the rib eye samples used in the study are listed in the Appendix. Marbling scores for the 23 samples ranged from 840 to 1080. Based on these scores, each sample was assigned a quality grade using the following scheme: Standard(less than 900), Select(900–990), Choice(1000–1200), and Prime(1300 and above). Five samples were assigned to Standard, 13 to Select, 5 to Choice, and none to Prime. The %fat values, determined using the n-hexane method, ranged from 1.46% to 7.72%. Figure 4.1 shows the relationship between subjectively determined marbling scores and chemically determined %fat values. As can be seen in this figure, there is almost a linear relationship between the two tissue features with good correlation( $\rho = 0.79$ ).



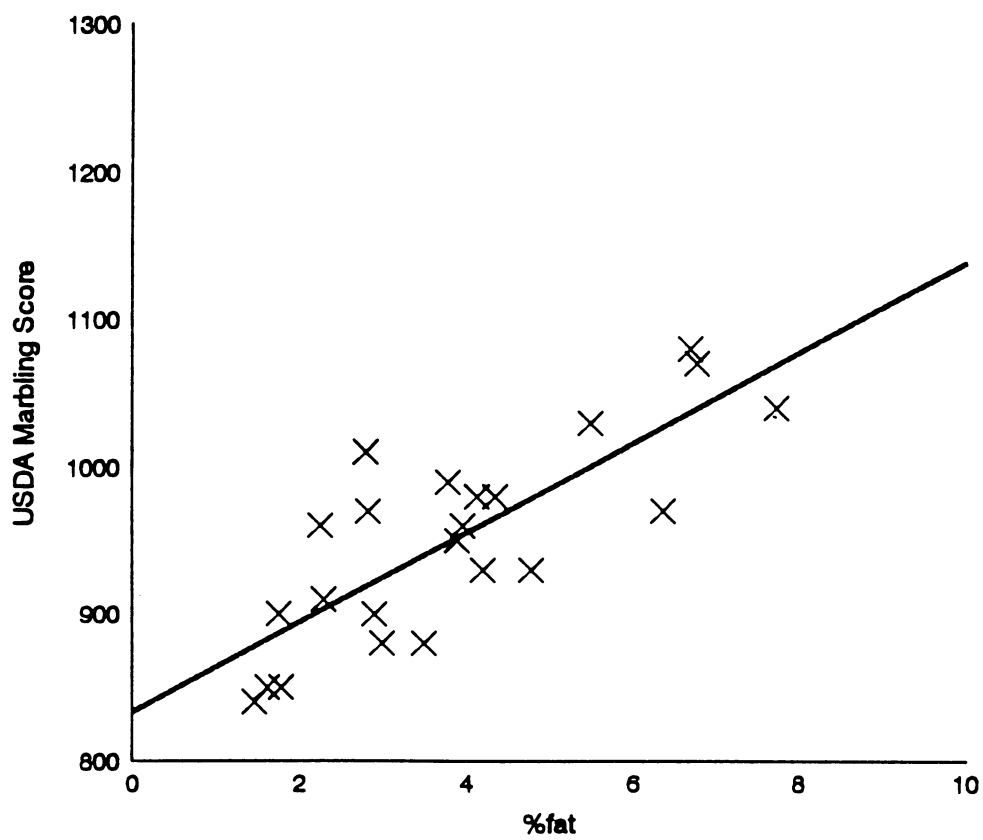


Figure 4.1: Correlation between subjectively determined marbling score and chemically determined %fat for the rib eye samples used in this study

## Correlation Analysis

The coefficients of correlation( $\rho$ ) between the textural features, marbling score, and %fat are presented in Table 4.1 (for the SGLDM method) and Table 4.2 (for the GLRLM method). The underlined values are significant to the 5% level. The first and second lines in the tables give the correlation of each of the features with marbling scores and %fat, respectively. The rest of the lines show the mutual correlation between the textural features.

### SGLDM based Features

Referring to Table 4.1, features such as Con, IDM, DV, DE, and IMC1 have good correlation with both marbling score and %fat. The feature SA, correlated very poorly with both the tissue features. Also, some features like ASM, Var, SV, SE, and Ent had better correlations with marbling score while others like Corr and IMC2 had better correlations with %fat. The mutual correlations between some of the textural features were very high which means that there is a lot of redundancy in the textural information captured by these features. Examples of strong correlations between features are given below with  $\rho$  values in brackets: Con and DV (0.99), Con and DE (0.98), IDM and DE (-0.97), and Var and SV (0.99).

### GLRLM based Features

Among the GLRLM based features, only SRE, LRE, GLD, RLD, and RP had significant correlations with the tissue features, marbling score and %fat (Table 4.2). Like the SGLDM based features, there was a general trend of strong mutual correlations between the features. Features like SRE and RLD ( $\rho = 0.99$ ),

Table 4.1: Mutual correlation coefficients between SGLDM based textural features and correlation with marbling score and %fat <sup>a</sup>

	marb	%fat	ASM	Con	Corr	Var	IDM	SA	SV	SE	Ent	DV	DE	IMC1	IMC2
marb	1.0	<u>0.79</u>	0.29	-0.51	0.30	-0.29	0.42	-0.06	-0.29	-0.22	-0.34	-0.54	-0.48	-0.31	0.13
%fat		1.0	0.12	<u>-0.43</u>	0.40	-0.12	0.34	0.02	-0.11	-0.03	-0.20	-0.47	-0.42	-0.42	0.30
ASM			1.0	-0.70	0.05	-0.74	0.85	-0.83	-0.74	-0.90	-0.96	-0.67	-0.76	-0.17	-0.32
Con				1.0	-0.54	<u>0.65</u>	-0.93	<u>0.62</u>	<u>0.64</u>	<u>0.62</u>	<u>0.84</u>	<u>0.99</u>	<u>0.98</u>	<u>0.58</u>	-0.16
Corr					1.0	0.27	0.51	0.10	0.29	0.26	-0.13	-0.56	-0.57	-0.97	0.90
Var						1.0	-0.59	0.80	0.99	0.93	0.83	0.63	0.61	-0.22	<u>0.63</u>
IDM							1.0	-0.73	-0.58	-0.68	-0.91	-0.91	-0.97	-0.60	0.13
SA								1.0	0.80	0.91	0.89	0.58	0.65	-0.12	<u>0.47</u>
SV									1.0	0.93	0.83	0.62	0.60	-0.23	0.64
SE										1.0	0.92	0.59	0.63	-0.16	<u>0.63</u>
Ent											1.0	0.81	0.87	0.23	0.27
DV												1.0	<u>0.97</u>	<u>0.58</u>	-0.17
DE													1.0	<u>0.63</u>	-0.19
IMC1														1.0	<u>1.0</u>
IMC2															<u>1.0</u>

<sup>a</sup>The textural features are: ASM = angular second moment, Con = contrast, Corr = correlation, Var = variance, IDM = inverse difference moment, SA = sum average, SV = sum variance, SE = sum entropy, Ent = entropy, DV = difference variance, DE = difference entropy, IMC1 & IMC2 = information measures of correlation.

Table 4.2: Mutual correlation coefficients between GLRLM based textural features and correlation with marbling score and %fat <sup>a</sup>

	marb	%fat	SRE	LRE	GLD	RLD	RP	LGRE	HGRE	SRLGE	SRHGE	LRHGE	LRIGE	LRIGE	LRIGE
marb	1.0	<u>0.79</u>	-0.41	0.39	0.13	-0.40	-0.40	0.04	0.06	0.01	0.06	-0.03	0.15		
%fat		1.0	-0.31	0.32	-0.06	-0.31	-0.31	-0.06	0.23	-0.09	0.10	0.08	0.03		
SRE			1.0	<u>-0.98</u>	<u>-0.58</u>	<u>0.99</u>	<u>0.99</u>	<u>-0.72</u>	<u>0.59</u>	<u>-0.67</u>	<u>0.68</u>	<u>0.13</u>	<u>-0.80</u>		
LRE				1.0	<u>0.54</u>	<u>-0.99</u>	<u>-0.99</u>	<u>0.70</u>	<u>-0.56</u>	<u>0.65</u>	<u>-0.65</u>	<u>-0.09</u>	<u>0.78</u>		
GLD					1.0	<u>-0.58</u>	<u>-0.57</u>	<u>0.85</u>	<u>-0.64</u>	<u>0.87</u>	<u>-0.77</u>	<u>-0.21</u>	<u>0.83</u>		
RLD						1.0	<u>0.99</u>	<u>-0.72</u>	<u>0.59</u>	<u>-0.67</u>	<u>0.69</u>	<u>0.12</u>	<u>-0.80</u>		
RP							1.0	<u>-0.71</u>	<u>0.58</u>	<u>-0.67</u>	<u>0.67</u>	<u>0.11</u>	<u>-0.79</u>		
LGRE								1.0	<u>-0.80</u>	<u>0.99</u>	<u>-0.94</u>	<u>-0.26</u>	<u>0.99</u>		
HGRE									1.0	<u>-0.79</u>	<u>0.82</u>	<u>0.22</u>	<u>-0.77</u>		
SRLGE										1.0	<u>-0.94</u>	<u>-0.27</u>	<u>0.97</u>		
SRHGE											1.0	<u>0.22</u>	<u>-0.90</u>		
LRHGE												1.0	<u>-0.28</u>		
LRIGE														1.0	

<sup>a</sup>The textural features are: SRE = short run emphasis, LRE = long run emphasis, GLD = grey level distribution, RLD = run length distribution, RP = run percentage, LGRE = low gray-level run emphasis, HGRE = high gray-level run emphasis, SRLGE = short run low gray-level emphasis, SRHGE = short run high gray-level emphasis, LRIGE = long run high gray-level emphasis, LRIGE = long run low gray-level emphasis.

LRE and RP (0.99), RLD and RP (0.99), and LGRE and SRLGE (0.99) represent similar textural information.

### **Regression Models**

Multiple regression methods were used to derive the optimum linear combination of features such that the regression models gave the maximum coefficient of determination ( $R^2$ ), the minimum root MSE (for adequacy of the model), and  $p$  value less than 0.05 (for significance of the model). The regression models for predicting marbling score and %fat using SGLDM and GLRLM based textural features are given in Figure 4.2. Figure 4.3 shows the prediction of %fat using SGLDM based features. There is a general trend among the regression models to over predict low values and under predict high values of %fat and marbling score. Increasing the data set to have a more even distribution may help to improve the predicting ability of the regression models.

	SGLDM Method	GLRLM Method
Marbling Score	# of features = 7 $R^2 = 0.57$ root MSE = 55 $p < 0.05$	# of features = 5 $R^2 = 0.51$ root MSE = 54 $p < 0.05$
%fat	# of features = 7 $R^2 = 0.60$ root MSE = 1.36 $p < 0.05$	# of features = 3 $R^2 = 0.36$ root MSE = 1.53 $p < 0.05$

Figure 4.2: Regression models for predicting marbling score and %fat using SGLDM and GLRLM based textural features

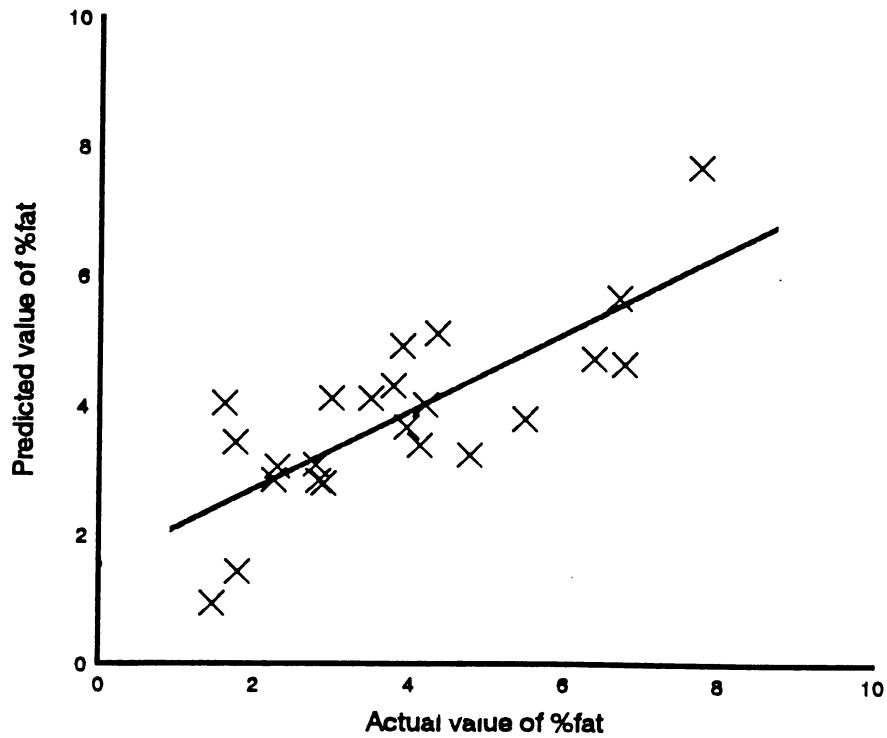


Figure 4.3: Predicted %fat versus actual %fat using SGLDM based textural features

## CHAPTER 5. CONCLUSION AND RECOMMENDATIONS FOR FUTURE RESEARCH

In this study, texture analysis of ultrasonic B-mode images using the SGLDM (spatial gray-level dependence matrices method) and GLRLM (gray-level run length matrices method) methods was investigated for predicting marbling in the rib eye muscle of live beef animals. Statistical analysis was used to develop regression models for predicting marbling score and %fat. Both tissue features were predicted with good accuracy. The SGLDM method gave better results than the GLRLM method for this application.

This study showed that B-mode image analysis is a promising alternative approach for predicting marbling and that it should be pursued in addition to A-mode signal analysis which has been studied for the past 10 years. Efforts are underway at Iowa State University to continue this study using a larger data set. Pattern recognition methods such as discriminant analysis and artificial neural networks can be used to develop a classification scheme based on the textural features for assigning quality grades. A feasibility study of applying neural networks for this application has already been done using features based on spectrum analysis of A-mode signals (Amin et al., 1992).

Since there was a lot of redundancy in the information represented by the textu-

ral features used in this study, other texture analysis methods such as Fourier power spectrum (FPS) and fractional brownian motion (Wu et al., 1992) can be investigated to see if there is any improvement in predicting marbling. Wu et al. (1992) have reported the better performance of fractional brownian motion model based feature set when compared to features based on the SGLDM and FPS methods for classification of ultrasonic liver images. Other features such as age and back-fat layer thickness can be used to improve the predicting ability of the above models.

The final goal is two fold: 1) to develop a robust scheme for objectively assigning quality grades to beef carcasses and 2) to accurately predict marbling in live beef animals so that they can be more effectively marketed.



**BIBLIOGRAPHY**

- Amin, V. R. 1989. Ultrasonic attenuation estimation for tissue characterization. M. S. Thesis. Iowa State University, Ames, Iowa.
- Amin, V. R. 1992. Tissue characterization by ultrasound: A study of tissue-mimicking materials and quantitative ultrasonics. Ph.D. Dissertation. Iowa State University, Ames, Iowa.
- Amin, V. R., V. J. Doerr, P. R. Arul, and D. L. Carlson. 1992. Application of neural network to ultrasound tissue characterization using backscattered signal parameters. *Proceedings of IEEE 1992 Nuclear Science Symposium and Medical Imaging Conference*. IEEE publications, New York.
- Brethour, J. R. 1990. Relationship of ultrasound speckle to marbling score in cattle. *Journal of Animal Science* 68:2603-2613.
- Chang, Y. K. 1991. Utility of the backscattered ultrasonic A-mode signal for the quantitative grading of beef. M. S. Thesis. Iowa State University.
- Chu, A., C. M. Sehgal, and J. F. Greenleaf. 1990. Use of gray value distribution of run lengths for texture analysis. *Pattern Recognition Letters* 11:415-420.
- Dasarathy, B. V., and E. B. Holder. 1991. Image characterizations based on joint gray level-run length distributions. *Pattern Recognition Letters* 12:497-502.
- Dasarathy, B. V. 1992. Texture based classification of cell imagery. *SPIE Medical Imaging VI: Image Processing* 1652:284-291.
- Edmonds, P. D., C. L. Mortensen, J. R. Hill, S. K. Holland, J. F. Jensen, P. Schattner, A. D. Valdes, R. H. Lee, and F. A. Marzoni. 1991. Ultrasound tissue characterization of breast biopsy specimens. *Ultrasonic Imaging* 13:162-185.

- Feleppa, E. J., and M. M. Yaremko. 1987. Ultrasonic tissue characterization for diagnosis and monitoring. *IEEE Engineering in Medicine and Biology Magazine* 6:18-26.
- Galloway, M. M. 1975. Texture analysis using gray level run lengths. *Computer Graphics and Image Processing* 4:172-179.
- Haralick, R. M., K. Shanmugam, and I. Dinstein. 1973. Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics* SMC-3:610-621.
- Haumschild, D. J., and D. L. Carlson. 1983. An ultrasonic Bragg scattering technique for the quantitative characterization of marbling in beef. *Ultrasonics* 21:226-233.
- Layer, G., I. Zuna, A. Lorenz, H. Zerban, U. Haberkorn, P. Bannasch, G. van Kaick, and U. R ath. 1990. Computerized ultrasound B-scan texture analysis of experimental fatty liver disease: Influence of total lipid content and fat deposit distribution. *Ultrasonic Imaging* 12:171-188.
- Lerski, R. A., M. J. Smith, P. Morley, E. Barnett, P. R. Mills, G. Watkinson, and R. N. M. MacSween. 1981. Discriminant analysis of ultrasonic liver data in diffuse alcoholic liver disease. *Ultrasonic Imaging* 3:164-172.
- Narayana, P. A., and J. Ophir. 1983. On the frequency dependence of attenuation in normal and fatty liver. *IEEE Transactions on Sonics and Ultrasonics* SU-30:379-383.
- Nicholas, D., D. K. Nassiri, P. Garbutt, and C. R. Hill. 1986. Tissue characterization from ultrasound B-scan data. *Ultrasound in Medicine and Biology* 12:135-143.
- Savell, J. W., R. H. Cross, and G. C. Smith. 1986. Percentage ether extractable fat and moisture content of beef longissimus muscle as related to USDA marbling score. *Journal of Food Science* 51:838-840.
- Shearer, S. A., and R. G. Holmes. 1990. Plant identification using color co-occurrence matrices. *Transactions of the ASAE* 33:2037-2044.

Widyaatmadja, L. B. 1991. Attenuation measurement of the backscattered ultrasonic A-mode signal for beef grading. M. S. Thesis. Iowa State University, Ames, Iowa.

Wu, C., Y. Chen, K. Hsieh 1992. Texture features for classification of ultrasonic liver images. *IEEE Transactions on Medical Imaging* 11:141-152.

**APPENDIX****Marbling Score and %Fat Values for Rib Eye Samples**

Sample-ID	Marbling Score	%Fat
047	840	1.46
051	980	4.12
053	930	4.19
060	1080	6.68
061	960	3.95
071	980	4.34
075	850	1.78
083	1070	6.76
087	1040	7.72
088	900	2.89
094	960	2.25
099	930	4.77
101	950	3.89
106	850	1.61
110	970	2.82
113	990	3.77
135	880	3.49
138	1030	5.48
157	970	6.36
171	880	2.99
174	1010	2.79
186	900	1.75
189	910	2.29

## C Programs for Computing Textural Features

```

/* Program for calculating textural features based on the spatial
** gray-level dependence matrices method.
**
** Author: P. R. Arul
**
**
*/

#include <stdio.h>
#include <math.h>
#define LX 80
#define LY 80
#define G 256
#define d 1
#define epsilon 0.000000001

extern void main1(int I[LX][LY],FILE *fp4);

gtsdm(int I[LY][LX]);
display(float **p);
mpmrow(float **p,float *q);
mpmcol(float **p,float *q);
funcpx_plus_y(float **p,float *q);
funcpx_minus_y(float **p,float *q);
float ang_second_mom(float **p);
float findcorr(float **p, float q[G], float r[G]);
float findcontrast(float **p);
float findvariance(float **p);
float inv_diff_mom(float **p);
float sum_avg(float q[2*G-1]);
float sum_entropy(float q[2*G-1]);
float sum_variance(float q[2*G-1], float SA);
float findentropy(float **p);
float diff_entropy(float q[G]);
float diff_variance(float q[G]);
float *info_M_corr(float **p,float q[G],float r[G],float EP);
float **matrix();

float **p0,**p45,**p90,**p135,px0[G],px45[G],px90[G],px135[G];

```

```

float py0[G],py45[G],py90[G],py135[G],px_p_y0[2*G+1],px_p_y45[2*G+1],
px_p_y90[2*G+1],px_p_y135[2*G+1];
float px_m_y0[G],px_m_y45[G],px_m_y90[G],px_m_y135[G];
float meu_x;
float EP1,EP2,EP3,EP4;
int glevel[G];
int level;
FILE *fp2,*fp4;

main(int argc, char *argv[])
{
    FILE *fp1;
    int I[LY][LX],i,j,k;
    float R0,R45,R90;
    float ASM1,ASM2,ASM3,ASM4,CON1,CON2,CON3,CON4;
    float CORR1,CORR2,CORR3,CORR4,VAR1,VAR2,VAR3,VAR4;
    float IDM1,IDM2,IDM3,IDM4,SA1,SA2,SA3,SA4;
    float SE1,SE2,SE3,SE4,SV1,SV2,SV3,SV4;
    float DE1,DE2,DE3,DE4,DV1,DV2,DV3,DV4;
    float *ptr1;
    float IN_CORR11,IN_CORR12,IN_CORR13,IN_CORR14,IN_CORR21,IN_CORR22,
    IN_CORR23,IN_CORR24;

    fp1 = fopen(argv[1],"r");
    for(i=0; i<LY; i++){
        for(j=0; j<LX; j++){
            fscanf(fp1,"%d",&I[i][j]);
        }
    }
    for(i=0; i<G; i++)
        glevel[i] = -1;
    for(i=0; i<LX; i++){
        for(j=0; j<LY; j++){
            glevel[I[i][j]]=I[i][j];
        }
    }

    for(k=0, level=0; k<G; k++)
        if(glevel[k]!=-1)
            glevel[level++] = glevel[k];

```

```

printf("no. of grey levels in the image= %d\n",level);

/* Allocate memory for spatial gray-level dependence matrices
p0 = matrix (0, level, 0, level);
p45 = matrix (0, level, 0, level);
p90 = matrix (0, level, 0, level);
p135 = matrix (0, level, 0, level);

for(i=0; i<level; i++)
  for(j=0; j<level; j++){
    p0[i][j] = p45[i][j] = 0;
    p90[i][j] = p135[i][j] = 0;
  }
printf("computing co-occurrence matrices.....");
gtsdm(I);
printf("done\n");
fp4 = fopen("tex_result","a");

/* Normalizing constants */
R0 = 2.0 * LY * (LX - d);
R45 = 2.0 * (LX - d) * (LY - d);
R90 = 2.0 * (LY - d) * LX;

for(i=0; i<level; i++)
  for(j=0; j<level; j++){
    p0[i][j] /= R0;
    p45[i][j] /= R45;
    p90[i][j] /= R90;
    p135[i][j] /= R45;
  }
fp2 = fopen("gtsdmatrix1","w");

printf("computing parameters.....");
mpmrow(p0,px0); mpmrow(p45,px45); mpmrow(p90,px90);
mpmrow(p135,px135);
mpmcol(p0,py0); mpmcol(p45,py45); mpmcol(p90,py90);
mpmcol(p135,py135);

funcpx_plus_y(p0,px_p_y0); funcpx_plus_y(p45,px_p_y45);
funcpx_plus_y(p90,px_p_y90); funcpx_plus_y(p135,px_p_y135);

```

```

funcpx_minus_y(p0,px_m_y0); funcpx_minus_y(p45,px_m_y45);
funcpx_minus_y(p90,px_m_y90); funcpx_minus_y(p135,px_m_y135);

ASM1 = ang_second_mom(p0); ASM2 = ang_second_mom(p45);
ASM3 = ang_second_mom(p90); ASM4 = ang_second_mom(p135);

CORR1 = findcorr(p0,px0,py0); CORR2 = findcorr(p45,px45,py45);
CORR3 = findcorr(p90,px90,py90); CORR4 = findcorr(p135,px135,py135);

CON1 = findcontrast(p0); CON2 = findcontrast(p45);
CON3 = findcontrast(p90); CON4 = findcontrast(p135);

VAR1 = findvariance(p0); VAR2 = findvariance(p45);
VAR3 = findvariance(p90); VAR4 = findvariance(p135);

IDM1 = inv_diff_mom(p0); IDM2 = inv_diff_mom(p45);
IDM3 = inv_diff_mom(p90); IDM4 = inv_diff_mom(p135);

SA1 = sum_avg(px_p_y0); SA2 = sum_avg(px_p_y45);
SA3 = sum_avg(px_p_y90); SA4 = sum_avg(px_p_y135);

SE1 = sum_entropy(px_p_y0); SE2 = sum_entropy(px_p_y45);
SE3 = sum_entropy(px_p_y90); SE4 = sum_entropy(px_p_y135);

SV1 = sum_variance(px_p_y0,SA1); SV2 = sum_variance(px_p_y45,SA2);
SV3 = sum_variance(px_p_y90,SA3); SV4 = sum_variance(px_p_y135,SA4);

EP1 = findentropy(p0); EP2 = findentropy(p45);
EP3 = findentropy(p90); EP4 = findentropy(p135);

DE1 = diff_entropy(px_m_y0); DE2 = diff_entropy(px_m_y45);
DE3 = diff_entropy(px_m_y90); DE4 = diff_entropy(px_m_y135);

DV1 = diff_variance(px_m_y0); DV2 = diff_variance(px_m_y45);
DV3 = diff_variance(px_m_y90); DV4 = diff_variance(px_m_y135);

ptr1 = info_M_corr(p0,px0,py0,EP1); IN_CORR11 = *ptr1++;
IN_CORR21 = *ptr1;
ptr1 = info_M_corr(p45,px45,py45,EP2); IN_CORR12 = *ptr1++;
IN_CORR22 = *ptr1;

```



```

ptr1 = info_M_corr(p90,px90,py90,EP3); IN_CORR13 = *ptr1++;
IN_CORR23 = *ptr1;
ptr1= info_M_corr(p135,px135,py135,EP4);IN_CORR14 = *ptr1++;
IN_CORR24 = *ptr1;

fprintf(fp4,"%s\n",argv[1]);

fprintf(fp4,"c1\t%11.5f\t%11.5f\t%11.5f\t%11.5f\t%11.5f\n",ASM1,
ASM2,ASM3,ASM4,(ASM1+ASM2+ASM3+ASM4)/4.0);
fprintf(fp4,"c2\t%11.5f\t%11.5f\t%11.5f\t%11.5f\t%11.5f\n",CON1,
CON2,CON3,CON4,(CON1+CON2+CON3+CON4)/4.0);
fprintf(fp4,"c3\t%11.5f\t%11.5f\t%11.5f\t%11.5f\t%11.5f\n",CORR1,
CORR2,CORR3,CORR4,(CORR1+CORR2+CORR3+CORR4)/4.0);
fprintf(fp4,"c4\t%11.5f\t%11.5f\t%11.5f\t%11.5f\t%11.5f\n",VAR1,
VAR2,VAR3,VAR4,(VAR1+VAR2+VAR3+VAR4)/4.0);
fprintf(fp4,"c5\t%11.5f\t%11.5f\t%11.5f\t%11.5f\t%11.5f\n",IDM1,
IDM2,IDM3,IDM4,(IDM1+IDM2+IDM3+IDM4)/4.0);
fprintf(fp4,"c6\t%11.5f\t%11.5f\t%11.5f\t%11.5f\t%11.5f\n",SA1,
SA2,SA3,SA4,(SA1+SA2+SA3+SA4)/4.0);
fprintf(fp4,"c7\t%11.5f\t%11.5f\t%11.5f\t%11.5f\t%11.5f\n",SV1,
SV2,SV3,SV4,(SV1+SV2+SV3+SV4)/4.0);
fprintf(fp4,"c8\t%11.5f\t%11.5f\t%11.5f\t%11.5f\t%11.5f\n",SE1,
SE2,SE3,SE4,(SE1+SE2+SE3+SE4)/4.0);
fprintf(fp4,"c9\t%11.5f\t%11.5f\t%11.5f\t%11.5f\t%11.5f\n",EP1,
EP2,EP3,EP4,(EP1+EP2+EP3+EP4)/4.0);
fprintf(fp4,"c10\t%11.5f\t%11.5f\t%11.5f\t%11.5f\t%11.5f\n",DV1,
DV2,DV3,DV4,(DV1+DV2+DV3+DV4)/4.0);
fprintf(fp4,"c11\t%11.5f\t%11.5f\t%11.5f\t%11.5f\t%11.5f\n",DE1,
DE2,DE3,DE4,(DE1+DE2+DE3+DE4)/4.0);
fprintf(fp4,"c12\t%11.5f\t%11.5f\t%11.5f\t%11.5f\t%11.5f\n",
IN_CORR11,IN_CORR12,IN_CORR13,IN_CORR14,
(IN_CORR11+IN_CORR12+IN_CORR13+IN_CORR14)/4.0);
fprintf(fp4,"c13\t%11.5f\t%11.5f\t%11.5f\t%11.5f\t%11.5f\n",
IN_CORR21,IN_CORR22,IN_CORR23,IN_CORR24,
(IN_CORR21+IN_CORR22+IN_CORR23+IN_CORR24)/4.0);

printf("done\n");
main1(I,fp4);
fclose(fp1);
fclose(fp2);

```

```

fclose(fp4);
}

/*FUNCTION TO FIND THE SPATIAL GRAY-LEVEL DEPENDENCE MATRICES
gtsdm(int I[][LY])
{
  int i,j,x,y,angle;

  for (i= 0; i< LX; i++)
    for (j = 0; j < LY; j++)
      for (x = 0, angle = 0; angle <= 135; angle += 45)
        {
while (glevel[x] != I[i][j])
  x++;
if (angle == 0 && j + d < LY)
{
  y = 0;
  while (glevel[y] != I[i][j + d])
    y++;
  p0[x][y]++;
  p0[y][x]++;
}
if (angle == 90 && i + d < LX)
{
  y = 0;
  while (glevel[y] != I[i + d][j])
    y++;
  p90[x][y]++;
  p90[y][x]++;
}
if (angle == 45 && i + d < LX && j - d >= 0)
{
  y = 0;
  while (glevel[y] != I[i + d][j - d])
    y++;
  p45[x][y]++;
  p45[y][x]++;
}
if (angle == 135 && i + d < LX && j + d < LY)
{

```

```

y = 0;
while (glevel[y] != I[i + d][j + d])
    y++;
p135[x][y]++;
p135[y][x]++;
}
    }
}

```

```

display(float **p)
{
    int i,j;
    for(i=0; i<level; i++){
        for(j=0; j<level; j++){
            fprintf(fp2," %f ",p[i][j]);
        }
        fprintf(fp2,"\n");
    }
    fprintf(fp2,"\n");
}

```

```

/* FUNCTION TO FIND THE MARGINAL PROB MATRIX BY SUMMING THE ROWS */
mpmrow(float **p,float *q)
{
    int i,j;
    float sum;
    for(i=0; i<level; i++){
        sum = 0;
        for(j=0; j<level; j++){
            sum += p[i][j];
        }
        *(q+i) = sum;
    }
}

```

```

/*FUNCTION TO FIND THE MARGINAL PROB MATRIX BY SUMMING THE COLUMNS */
mpmcol(float **p,float *q)
{
    int i,j;
    float sum;

```

```

for(i=0; i<level; i++){
    sum = 0;
    for(j=0; j<level; j++){
        sum += p[j][i];
    }
    *(q+i) = sum;
}
}

/*FUNCTION TO FIND P X+Y */
funcpx_plus_y(float **p,float *q)
{
    int i,j,k;
    float sum;
    for(k=0; k<(2*level-1); k++){
        sum = 0;
        for(i=0; i<level; i++){
            for(j=0; j<level; j++){
                if((i+j) == k)
                    sum += p[i][j];
            }
        }
        *(q+k+2) = sum;
    }
}

/*FUNCTION TO FIND P X-Y */
funcpx_minus_y(float **p,float *q)
{
    int i,j,k;
    float sum;
    for(k=0; k<level; k++){
        sum = 0;
        for(i=0; i<level; i++){
            for(j=0; j<level; j++){
                if(abs(i-j) == k)
                    sum += p[i][j];
            }
        }
        *(q+k) = sum;
    }
}

```

```

}
}

/* FUNCTION TO FIND THE ANGULAR SECOND MOMENT */
float ang_second_mom(float **p)
{
    int i,j;
    float sum=0;
    for(i=0; i<level; i++){
        for(j=0; j<level; j++){
            sum += p[i][j]*p[i][j];
        }
    }
    return sum;
}

/* FUNCTION TO FIND CORRELATION */
float findcorr(float **p, float q[G], float r[G])
{
    int i,j;
    float sum=0,meu_y,sd_x,sd_y,corr;

    for(i=0; i<level; i++){
        sum += i*q[i];
    }
    meu_x = sum;
    sum = 0;
    for(j=0; j<level; j++){
        sum += j*r[j];
    }
    meu_y = sum;
    sum = 0;
    for(i=0; i<level; i++){
        sum += (i- meu_x)*(i - meu_x)*q[i];
    }
    sd_x = sqrt(sum);
    sum = 0;
    for(j=0; j<level; j++){
        sum += (j - meu_y)*(j - meu_y)*r[j];
    }
}

```

```

sd_y = sqrt(sum);
sum = 0;
for(i=0; i<level; i++){
    for(j=0; j<level; j++){
        sum += i*j*p[i][j];
    }
}
corr = (sum - meu_x*meu_y)/(sd_x*sd_y);
return corr;
}

```

```

/* FUNCTION TO FIND CONTRAST
float findcontrast(float **p)
{

```

```

    int i,j,n;
    float sum,sum1=0;
    for(n=0; n<level; n++){
        sum = 0;
        for(i=0; i<level; i++){
            for(j=0; j<level; j++){
if(abs(i-j) == n)
                sum += p[i][j];
            }
        }
        sum1 += n*n*sum;
    }
    return sum1;
}

```

```

/* FUNCTION TO FIND VARIANCE                                     */
float findvariance(float **p)
{
    int i,j;
    float sum=0;
    for(i=1; i<level; i++){
        for(j=1; j<level; j++){
            sum += p[i][j]*(i - meu_x)*(i - meu_x);
        }
    }
    return sum;
}

```

```

}

/* FUNCTION TO FIND THE INVERSE DIFFERENCE MOMENT
float inv_diff_mom(float **p)
{
    int i,j;
    float sum=0;
    for(i=0; i<level; i++){
        for(j=0; j<level; j++){
            sum += p[i][j]/(1+(i-j)*(i-j));
        }
    }
    return sum;
}

/* FUNCTION TO FIND THE SUM AVERAGE                                     */
float sum_avg(float q[2*G+1])
{
    int i;
    float sum=0;
    for(i=2; i<=(2*level); i++){
        sum += i*q[i];
    }
    return sum;
}

/* FUNCTION TO FIND SUM ENTROPY                                       */
float sum_entropy(float q[2*G+1])
{
    int i;
    float sum=0;
    for(i=2; i<=(2*level); i++){
        sum += q[i] * log10(q[i] + epsilon);
    }
    return -sum;
}

/* FUNCTION TO FIND SUM VARIANCE                                       */
float sum_variance(float q[2*G+1], float SA)
{
    int i;
    float sum=0;

```

```

    for(i=2; i<=(2*level); i++){
        sum += (i - SA)*(i - SA)*q[i];
    }
    return sum;
}
/* FUNCTION TO FIND ENTROPY                                     */
float findentropy(float **p)
{
    int i,j;
    float sum=0;
    for(i=0; i<level; i++){
        for(j=0; j<level; j++){
            sum += p[i][j]*log10(p[i][j] + epsilon);
        }
    }
    return -sum;
}
/* FUNCTION TO FIND DIFFERENCE ENTROPY                         */
float diff_entropy(float q[G])
{
    int i;
    float sum=0;
    for(i=0; i<level; i++){
        sum += q[i]*log10(q[i] + epsilon);
    }
    return -sum;
}
/* FUNCTION TO FIND DIFFERENCE VARIANCE                       */
float diff_variance(float q[G])
{
    int i;
    float sum=0,DA=0;
    for(i=0; i<level; i++){
        DA += i*q[i];
    }
    for(i=0; i<level; i++){
        sum += (i - DA)*(i - DA)*q[i];
    }
    return sum;
}

```



```

/* FUNCTION TO FIND INFORMATION MEASURES OF CORRELATION */
float *info_M_corr(float **p,float q[G],float r[G],float EP)
{
  int i,j;
  float sum=0,sum1=0,sum2=0,sum3=0,HX,HY,HXY1,HXY2,max,a[2];
  for(i=0; i<level; i++){
    sum += q[i]*log10(q[i] + epsilon);
    sum1 += r[i]*log10(r[i] + epsilon);
    for(j=0; j<level; j++){
      sum2 += p[i][j]*log10(q[i]*r[j] + epsilon);
      sum3 += q[i]*r[j]*log10(q[i]*r[j] + epsilon);
    }
  }
  HX = -sum; HY = -sum1; HXY1 = -sum2; HXY2 = -sum3;
  max = (HX > HY) ? HX:HY;
  a[0] = (EP - HXY1)/max;
  a[1] = sqrt(1 - exp(-2.0*(HXY2 - EP)));
  return a;
}

float **matrix (nrl, nrh, ncl, nch)
  int nrl, nrh, ncl, nch;

/* Allocates a float matrix with range [nrl..nrh][ncl..nch] */
{
  int i;
  float **m;
  /* allocate pointers to rows */
  m = (float **) malloc ((unsigned) (nrh - nrl + 1) * sizeof
(float *));
  if (!m)
    fprintf (stderr, "memory allocation failure"), exit (1);
  m -= ncl;
  /* allocate rows and set pointers to them */
  for (i = nrl; i <= nrh; i++)
  {
    m[i] = (float *) malloc ((unsigned) (nch - ncl + 1) * sizeof
(float));
    if (!m[i])
      fprintf (stderr, "memory allocation failure"), exit (2);
  }
}

```

```
    m[i] -= ncl;
}
/* return pointer to array of pointers to rows
return m;
}
```

```

/* Program for calculating textural features based on the gray-level
** run length matrices.
**
** Author: P. R. Arul
**
**
**
*/

#include <stdio.h>
#define N 80
#define G 256
#define LX 80
#define LY 80

runlen(int I[LX][LY]);
display1(float R[G][N]);
float compute_Nr(float R[G][N]);
float sre(float R[G][N],float Nr);
float lre(float R[G][N],float Nr);
float gld(float R[G][N],float Nr);
float rld(float R[G][N],float Nr);
float lgre(float R[G][N],float Nr);
float hgre(float R[G][N],float Nr);
float srlge(float R[G][N],float Nr);
float srhge(float R[G][N],float Nr);
float lrhge(float R[G][N],float Nr);
float lrlge(float R[G][N],float Nr);

float R0[G][N],R45[G][N],R90[G][N],R135[G][N];
FILE *fp3;

main1(int I[][LY],FILE *fp4)
{
    int i,j;
    char ch1[25];
    float Nr1,Nr2,Nr3,Nr4;
    float SRE1,SRE2,SRE3,SRE4,LRE1,LRE2,LRE3,LRE4,GLD1,GLD2,GLD3,GLD4;
    float RLD1,RLD2,RLD3,RLD4,RPC1,RPC2,RPC3,RPC4;
    float LGRE1,LGRE2,LGRE3,LGRE4,HGRE1,HGRE2,HGRE3,HGRE4;

```

```

float SRLGE1,SRLGE2,SRLGE3,SRLGE4,SRHGE1,SRHGE2,SRHGE3,SRHGE4;
float LRHGE1,LRHGE2,LRHGE3,LRHGE4,LRLGE1,LRLGE2,LRLGE3,LRLGE4;

/*printf("name the image file\n");
scanf("%25s",ch1);
fp1 = fopen(ch1,"r");
for(i=0; i<LX; i++){
    for(j=0; j<LY; j++){
        fscanf(fp1,"%d",&I[i][j]);
    }
}*/
for(i=0; i<G; i++)
    for(j=0; j<N; j++){
        R0[i][j] = R45[i][j] = 0;
        R90[i][j] = R135[i][j] = 0;
    }
printf("computing run length matrices.....");
runlen(I);
printf("done\n");

fp3 = fopen("runlenmatrix1","w");
printf("computing run length parameters.....");

Nr1 = compute_Nr(R0);Nr2 = compute_Nr(R45);Nr3 = compute_Nr(R90);
Nr4 = compute_Nr(R135);
SRE1=sre(R0,Nr1);SRE2=sre(R45,Nr2);SRE3=sre(R90,Nr3);
SRE4=sre(R135,Nr4);
LRE1=lre(R0,Nr1);LRE2=lre(R45,Nr2);LRE3=lre(R90,Nr3);
LRE4=lre(R135,Nr4);
GLD1=gld(R0,Nr1);GLD2=gld(R45,Nr2);GLD3=gld(R90,Nr3);
GLD4=gld(R135,Nr4);
RLD1=rld(R0,Nr1);RLD2=rld(R45,Nr2);RLD3=rld(R90,Nr3);
RLD4=rld(R135,Nr4);
RPC1=Nr1/(LX*LY);RPC2=Nr2/(LX*LY);RPC3=Nr3/(LX*LY);RPC4=Nr4/(LX*LY);

LGRE1=lgre(R0,Nr1);LGRE2=lgre(R45,Nr2);LGRE3=lgre(R90,Nr3);
LGRE4=lgre(R135,Nr4);
HGRE1=hgre(R0,Nr1);HGRE2=hgre(R45,Nr2);HGRE3=hgre(R90,Nr3);
HGRE4=hgre(R135,Nr4);

```

```

    SRLGE1=srlge(R0,Nr1);SRLGE2=srlge(R45,Nr2);SRLGE3=srlge(R90,Nr3);
    SRLGE4=srlge(R135,Nr4);
    SRHGE1=srhge(R0,Nr1);SRHGE2=srhge(R45,Nr2);SRHGE3=srhge(R90,Nr3);
    SRHGE4=srhge(R135,Nr4);
    LRHGE1=lrhge(R0,Nr1);LRHGE2=lrhge(R45,Nr2);LRHGE3=lrhge(R90,Nr3);
    LRHGE4=lrhge(R135,Nr4);
    LRLGE1=lrlge(R0,Nr1);LRLGE2=lrlge(R45,Nr2);LRLGE3=lrlge(R90,Nr3);
    LRLGE4=lrlge(R135,Nr4);
    printf("done\n");

    fprintf(fp4,"r1\t%11.5f\t%11.5f\t%11.5f\t%11.5f\t%11.5f\n",SRE1,
    SRE2,SRE3,SRE4,(SRE1+SRE2+SRE3+SRE4)/4.0);
    fprintf(fp4,"r2\t%11.5f\t%11.5f\t%11.5f\t%11.5f\t%11.5f\n",LRE1,
    LRE2,LRE3,LRE4,(LRE1+LRE2+LRE3+LRE4)/4.0);
    fprintf(fp4,"r3\t%11.5f\t%11.5f\t%11.5f\t%11.5f\t%11.5f\n",GLD1,
    GLD2,GLD3,GLD4,(GLD1+GLD2+GLD3+GLD4)/4.0);
    fprintf(fp4,"r4\t%11.5f\t%11.5f\t%11.5f\t%11.5f\t%11.5f\n",RLD1,
    RLD2,RLD3,RLD4,(RLD1+RLD2+RLD3+RLD4)/4.0);
    fprintf(fp4,"r5\t%11.5f\t%11.5f\t%11.5f\t%11.5f\t%11.5f\n",RPC1,
    RPC2,RPC3,RPC4,(RPC1+RPC2+RPC3+RPC4)/4.0);
    fprintf(fp4,"r6\t%11.5f\t%11.5f\t%11.5f\t%11.5f\t%11.5f\n",LGRE1,
    LGRE2,LGRE3,LGRE4,(LGRE1+LGRE2+LGRE3+LGRE4)/4.0);
    fprintf(fp4,"r7\t%11.5f\t%11.5f\t%11.5f\t%11.5f\t%11.5f\n",HGRE1,
    HGRE2,HGRE3,HGRE4,(HGRE1+HGRE2+HGRE3+HGRE4)/4.0);
    fprintf(fp4,"r8\t%11.5f\t%11.5f\t%11.5f\t%11.5f\t%11.5f\n",SRLGE1,
    SRLGE2,SRLGE3,SRLGE4,(SRLGE1+SRLGE2+SRLGE3+SRLGE4)/4.0);
    fprintf(fp4,"r9\t%11.5f\t%11.5f\t%11.5f\t%11.5f\t%11.5f\n",SRHGE1,
    SRHGE2,SRHGE3,SRHGE4,(SRHGE1+SRHGE2+SRHGE3+SRHGE4)/4.0);
    fprintf(fp4,"r10\t%11.5f\t%11.5f\t%11.5f\t%11.5f\t%11.5f\n",LRHGE1,
    LRHGE2,LRHGE3,LRHGE4,(LRHGE1+LRHGE2+LRHGE3+LRHGE4)/4.0);
    fprintf(fp4,"r11\t%11.5f\t%11.5f\t%11.5f\t%11.5f\t%11.5f\n",LRLGE1,
    LRLGE2,LRLGE3,LRLGE4,(LRLGE1+LRLGE2+LRLGE3+LRLGE4)/4.0);

    fprintf(fp4,"\n");
    fclose(fp3);
}

/*function to compute the gray-level run length matrices */
runlen(int I[][LY])
{

```

```

int i,j,k,gl,rl,a,count=0;
for(gl=0; gl<G; gl++){
    for(i=0; i<LX; i++){
        for(j=0; j<LY; j++){
            if(I[i][j] == gl){
k=0;
while(j+k<LY && I[i][j+k] == gl){
    k++;
}
        j+=k;
        R0[gl][k]+=1;
    }
}
    for(i=0; i<LX; i++){
        for(j=0; j<LY; j++){
            if(I[i][j] == gl){
k=0;
if(i==0 || j==LY-1 || I[i-1][j+1]!=gl){
    while(i+k<LX && j-k>=0 && I[i+k][j-k] == gl){
        k++;
    }
    R45[gl][k]+=1;
}
        }
}
    for(j=0; j<LY; j++){
        for(i=0; i<LX; i++){
            if(I[i][j] == gl){
k=0;
while(i+k<LX && I[i+k][j] == gl){
    k++;
}
        i+=k;
        R90[gl][k]+=1;
    }
}
    for(i=0; i<LX; i++){

```



```

/* FUNCTION TO COMPUTE SHORT RUN EMPHASIS (SRE)                               */
float sre(float R[][N],float Nr)
{
    int i,j;
    float sum=0;
    for(i=1; i<G; i++){
        for(j=1; j<N; j++){
            sum += R[i][j]/(j*j);
        }
    }
    return sum/Nr;
}

```

```

/* FUNCTION TO COMPUTE LONG RUN EMPHASIS (LRE)                               */
float lre(float R[][N],float Nr)
{
    int i,j;
    float sum=0;
    for(i=1; i<G; i++){
        for(j=1; j<N; j++){
            sum += R[i][j]*(j*j);
        }
    }
    return sum/Nr;
}

```

```

/* FUNCTION TO COMPUTE GREY LEVEL DISTRIBUTION (GLD)                       */
float gld(float R[][N],float Nr)
{
    int i,j;
    float sum=0,sum1=0;
    for(i=1; i<G; i++){
        sum=0;
        for(j=1; j<N; j++){
            sum += R[i][j];
        }
        sum1 += sum*sum;
    }
    return sum1/Nr;
}

```



```

/* FUNCTION TO COMPUTE RUN LENGTH DISTRIBUTION (RLD)          */
float rld(float R[][N],float Nr)                               */
{
    int i,j;
    float sum=0,sum1=0;
    for(j=1; j<N; j++){
        sum=0;
        for(i=1; i<G; i++){
            sum += R[i][j];
        }
        sum1 += sum*sum;
    }
    return sum1/Nr;
}

/* FUNCTION TO COMPUTE LOW GRAY-LEVEL RUN EMPHASIS (LGRE)    */
float lgre(float R[][N],float Nr)                              */
{
    int i,j;
    float sum=0;
    for(i=1; i<G; i++){
        for(j=1; j<N; j++){
            sum += R[i][j]/(i*i);
        }
    }
    return sum/Nr;
}

/* FUNCTION TO COMPUTE HIGH GRAY-LEVEL RUN EMPHASIS (HGRE)  */
float hgre(float R[][N],float Nr)                              */
{
    int i,j;
    float sum=0;
    for(i=1; i<G; i++){
        for(j=1; j<N; j++){
            sum += R[i][j]*(i*i);
        }
    }
    return sum/Nr;
}

```

```

}
```

```

/* FUNCTION TO COMPUTE SHORT RUN LOW GRAY-LEVEL EMPHASIS (SRLGE) */
```

```
float srlge(float R[][N],float Nr)
```

```

{
  int i,j;
  float sum=0;
  for(i=1; i<G; i++){
    for(j=1; j<N; j++){
      sum += R[i][j]/(i*i*j*j);
    }
  }
  return sum/Nr;
}

```

```

/* FUNCTION TO COMPUTE SHORT RUN HIGH GRAY-LEVEL EMPHASIS (SRHGE) */
```

```
float srhge(float R[][N],float Nr)
```

```

{
  int i,j;
  float sum=0;
  for(i=0; i<G; i++){
    for(j=1; j<N; j++){
      sum += (i*i*R[i][j])/(j*j);
    }
  }
  return sum/Nr;
}

```

```

/* FUNCTION TO COMPUTE LONG RUN HIGH GRAY-LEVEL EMPHASIS (LRHGE) */
```

```
float lrhge(float R[][N],float Nr)
```

```

{
  int i,j;
  float sum=0;
  for(i=0; i<G; i++){
    for(j=0; j<N; j++){
      sum += i*i*j*j*R[i][j];
    }
  }
  return sum/Nr;
}

```

```
/* FUNCTION TO COMPUTE LONG RUN LOW GRAY-LEVEL EMPHASIS (LRLGE)
float lrlge(float R[][N],float Nr)
{
    int i,j;
    float sum=0;
    for(i=1; i<G; i++){
        for(j=0; j<N; j++){
            sum += (j*j*R[i][j])/(i*i);
        }
    }
    return sum/Nr;
}
```

## ACKNOWLEDGEMENTS

I would like to thank my major professor, Dr. David Carlson, for giving me the opportunity to conduct research under his guidance. I would also like to thank Dr. Mary Helen Greer, Dr. Curran Swift, and Dr. Tenkasi Ramabadran for serving on my program of study committee. I am grateful to Dr. Mary Helen Greer and the Biomedical Engineering Department for the administrative and financial support. Special thanks are due to Viren Amin for helping me in my research and encouraging me to publish papers. I would like to thank the staff at the Animal Science Department and Animal Science Extension for providing the images that were used in my research. The help of Ms. Margie Carter of Image Analysis Facility, in using the PV-WAVE software package is appreciated.

I am thankful to my grandparents, parents, brother, and sisters for their encouragement and support. I am also thankful to all my friends at ISU for making my stay here a memorable one.

263-689