Optical Sensing as a Knowledge-Based System Input to Determine Flour Mill Break Stream Quality

by

Keith William Hefty

B.S., Kansas State University, 1988

A THESIS

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Electrical Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1989

Approved by:

Eddie R. Fowler

Major Professor

Table of Contents

List of Figures

List of Tables

Acknowledgments

I am extremely grateful to Dr. Eddie Fowler, my major professor, for his guidance while I was conducting this research. I would like to express my sincere thanks to the other members of my committee Dr. Satish Chandra and Dr. Okkyung Chung.

My appreciation is extended to the United States Department of Agriculture Grain Marketing Research Laboratory for funding of the research. Also, I would like to acknowledge the efforts of Mr. Ron Hightower, a fellow graduate student, who tutored me in the use of the Grinnell/VAX image processing system. Thanks to Dr. Elieser Posner, Dr. Dale Eustace and Mr. Steve Curran for their advice and cooperation in the collection of samples from the pilot mill.

I would also like to thank my parents and my brother and sisters for their love and support. Finally, I thank God for giving me the talents of learning and the opportunity to share what I have learned with others.

Chapter 1

1.0 Introduction

Recent advancements in programmable logic controller technology and sensor technology, which provides input to the controller, have improved the way flour is milled. This thesis proposes the next step in flour mill automation, computer based control of the milling process using artificial intelligence technology. Specifically, this research focuses on the use of image processing technology to detect features in flour mill break stock. The information obtained from the image analysis serves as input data to a knowledge-based control system that controls the width of the break roll gaps.

1.1 Research Motivation

In current milling practice the break roll gaps are initially set using information primarily received while the miller visually examines the stock as it exits the break rolls. Fine tuning adjustments usually occur after the miller has received some feedback on the quality of flour the mill is producing. The proposed roll gap control system operates on the assumption that break stock quality directly affects the finished product quality. This system optimizes the break roll gap settings based on break stock quality alone.

Benefits derived from a roll gap control system of this type are increased finished product consistency and maintenance of optimal extraction rates. The roll gap control system constantly monitors the operation of the break rolls. Changes in the mill mix are detected as soon as they occur. This allows making adjustments in response to variations in the mill mix in a shorter period of time. Maintaining the consistency of the breaking process in this fashion allows for more consistent product to flow to other parts of the mill and ultimately leads to increases in the consistency of the flour being produced. Wingfield [1] has stated the importance of maintaining a consistent flow of material into each machine in a mill for mill control.

Clegg [2] has shown that each one percent improvement in the amount of flour produced by a typical 7500 hundredweight mill will result in an annual income increase of $110,000. Therefore, a minor increase in flour value, i.e., quality, quantity, and/or consistency, adds up to a significant monetary gain. Since a roll gap control system improves the value of the flour, it is asserted that this project is worthwhile.

1.2 Research Goals

The main goal of this research is to determine if detection of break stream quality is feasible by using image processing technology. A secondary goal is to find which image processing techniques are the best at differentiating features that indicate quality in break stocks. Chapter 2 of this thesis describes the research effort in this area. Additionally, chapter 3 discusses how the results of the research can be used to construct a knowledge-based system that controls break roll gaps. Chapter 4 summarizes the conclusions derived from this research.

1.3 Previous Work

The only known research conducted in the area of mill control by detecting intermediate mill stream quality was reported by Posner and Wetzel [3]. They used near infrared reflectance (NIR) technology to measure the cellulose content of a stream that feeds a purifier. This information was used to control the purifier operation. Unfortunately Kansas State University's Department of Grain Science and Industry is one of the few places in the world where non-proprietary research in milling technology is conducted. Little literature is available which reports research conducted in this area by private industry.

1.4 General Research Approach

Ideally, the use of a knowledge-based control system to control the break system can be extended to cover the whole mill [4]. This system would control the cleaning, blending, tempering, and milling of the wheat. Likewise, the blending of flours and any additives would also be controlled by the knowledge-based system.

To develop such a system requires: 1) identification of knowledge used for completing each step in the milling process; 2) finding out the best way to automatically acquire this information; 3) encoding of the heuristics, used to determine the desired machinery settings, into a knowledge base; and 4) the design and implementation of advanced control mechanisms to change the actual machine settings. The research and development of such a control system is impractical for incorporating in one thesis.

A practical approach is first to research and develop the control theory for a single system within the milling process, such as the break system. Then, develop the input sensing equipment needed and next, construct a prototype knowledge-based system for it. After the break system knowledge based system has evolved through a series of design or redesign changes and achieved a desired performance level, it can be expanded gradually to cover other sub-systems of the milling process. The research presented here represents the first step in such an effort.

1.5 Research Obstacles

The Grinnell/VAX image processing system in the Kansas State University (KSU) Electrical and Computer Engineering (EECE) Department was used to accomplish the image processing tasks in this research. This system lacked the appropriate software for this research. The author had to become familiar with the operation of the Grinnell/VAX system and writing image processing software. This knowledge was gained through tutoring by a fellow graduate student. The tutoring included how to operate the Grinnell/VAX system, write programs to control the Grinnell/VAX system, and employ basic image processing techniques.

The failure of the Grinnell system hardware on two separate occasions possibly changed its operating characteristics enough to invalidate correlation results when comparing data obtained before and after the failures.

Chapter 2

2.0 Perception of Mill Stream Characteristics.

The miller uses all of the human senses, i.e., seeing, feeling, hearing, smelling, and tasting, to perceive the characteristics and conditions of the entire mill stream while adjusting it for optimum operation. Below is a description of how the miller uses touch and sight to properly set the break system.

After setting the feed gate properly the miller checks for evenness of grind. Evenness of grind is primarily sensed by two different methods. First, the miller uses touch to sense the warmth of the stock at both ends of the rolls. Second, the miller takes a sample of the stock from both ends of the rolls and visually inspects the sample for the amount of free endosperm and the size of the bran flakes.

Once the rolls are adjusted to grind evenly, the miller again uses the senses of touch and sight to set the final gap of the rolls. Visually the miller can determine the closeness of the grind by judging how ragged or shredded the edges of the bran flakes appear, the presence of any bran specks, and the amount of fine particles produced. Touch is used to sense the sharpness of the endosperm particles and the warmth of the stock. The miller combines all of these cues to adjust the roll gap. In addition, a test sifter is normally used to precisely set the roll gap for a specific break release.

2.1 Optical Sensing of Mill Stream Characteristics

This chapter of the thesis focuses on optically determining the quality of break stock using the same or similar visual cues as used by the miller. This procedure requires the use of a monochrome camera to obtain a digitized picture of the stock. The picture digitization is followed by image processing techniques to determine the characteristics of the break stock. This image processing quantifies the amount of free endosperm, the raggedness of bran edges, the size of bran particles, etc.

The suggested procedure to obtain a digitized image of the stock is to place a camera under each set of rolls in the break flow. Since the stock is moving at a high rate of speed as it leaves the rolls, high speed photography technology will be utilized to freeze the photograph of the stock as it moves past the camera.

For the research presented here, high speed photography was not available. Therefore the following procedures were used. Samples of each break stream were taken while the mill was running. Individual samples were placed, one at a time, statically in front of the monochrome camera. The stock collection process required catching the sample underneath the rolls, transporting it to the camera room, and placing the stock in front of the camera. (It is believed that noise was introduced into the test data using this collection and transportation procedure, because the representative nature of the sample was destroyed during the procedure.)

Once the monochrome digitized picture of the mill stock is obtained, simple image processing techniques are used to isolate individual bran and endosperm particles in the picture. Calculations are performed on these areas of bran and endosperm particles to determine: 1) the number of particles in the picture (each particle being defined by a contiguous set of pixels); 2) the average size of the particles (measured by averaging the number of pixels that make up each particle); 3) the average gray level of the pixels within each particle; and 4) the ratio of total area of endosperm to bran.

The above described stock collections procedures and image processing techniques do not indicate evenness of grind or proper adjustment of the feed gate.

The number of particles found serves as a test of how well the sample is laid out in front of the camera and how well the software isolates bran and endosperm particles in the picture. The size of the particles indicates severity of grind and the roll gap setting. The average gray level of the particles indicates what layers of the wheat berry are being exposed, but only if the camera is sensitive enough to small changes in light intensity. The ratio of total area of endosperm to bran indicates the amount of free endosperm being released.

2.2 Imaging Processing Technique Development

The software for this project was developed over an eight month period starting in March 1988. The new code was written in VAX Fortran 77. Fortran 77 was used because the subroutines that control the Grinnell hardware are written in it.

Original code had to be developed because the KSU EECE Department resident application program which controls the Grinnell, called IMAGER, works only with 64 by 64 pixel images. In this research, 256 by 256 pixel images were used. The KSU EECE Department has a resident set of four library subroutines [3], which work on any size picture, that initialize the Grinnell, store and retrieve images from the disk, and display and grab images from the Grinnell. These subroutines have been used during the first phase of the code development. During this phase, programs were written to test or exercise the library subroutines. These programs form the first 4 options of the newly developed program.

The resulting program is called ISOAREA and has gone through four major revisions with the final version located in Appendix A. The program is driven by a single menu which contains 12 options. A brief overview of these option is presented with a detailed description of each option following.

Five options allow for movement of the digital images among the Grinnell digitizer, the VAX memory, and the VAX disk. Three options allow the user to perform the standard image processing functions of threshold, histogram, and histogram equalization. One option controls the Grinnell digitizer, enabling it which allows the user to take a picture. After the picture is taken, this option disenables the Grinnell digitizer. Two other options have been programmed specifically for this project. The first takes an image in memory and, based on gray levels, isolates the bran or endosperm particles in the picture. The second option computes the size and average gray level of each area isolated by the first option. A final option terminates the program and returns control to the VAX/VMS operating system.

Each option of the first phase of the software development is now discussed in detail. Option 1 grabs an image from the Grinnell and reads it into the VAX memory. The image is taken from the center of the 512 by 512 Grinnell screen. The lower left hand coordinates, in Grinnell screen coordinates, of the image grabbed are (128, 128). The image can be read into either one of the two image buffers in memory. The ISOAREA program contains two 256 by 256 arrays of buffers for holding image data. Whenever an operation is executed which alters the appearance of the original image, the resultant image is placed in the alternate buffer leaving the original image unchanged.

Option 2 displays a digitized image in one of the two image buffers to the Grinnell screen. The user is prompted for the lower left coordinates of the image. By choosing the coordinates (0,0), (0, 256), (256,0), and (256, 256), four images can be displayed simultaneously. Use of coordinates (128, 128) center the image on the Grinnell screen.

Option 3 saves an image from one of the two image buffers to the disk. The user is prompted for a filename. The program automatically appends the extension ".dat" to the filename.

Option 4 loads an image from disk into memory. Again the image can be read into either buffer and the user is prompted for a filename with an extension of ".dat" assumed.

The second phase of software development added the 5th through 7th options to the menu. Option 5 turns on the Grinnell digitizer upon its selection from the menu. When the user presses the return key, the digitizer is disenabled causing the picture to "freeze" and the image data is stored in the Grinnell display buffer.

Option 6 allows the user to compute the histogram of an image in the VAX memory. To display the histogram on the VAX terminal screen the number of gray levels is reduced from 256 to 64 by averaging the number of pixels in 4 adjacent gray values. Then the histogram is normalized to a maximum value of 24. The user is given the option of saving the histogram to a file for output to a printer at a later time.

Option 7 thresholds an image in the VAX memory. After prompting the user for the threshold value, this option turns all pixels, whose gray level is less then the threshold, black and all pixels, whose gray level is greater than the threshold, white. This option leaves the original image unchanged in its original buffer and places the threshold image in the alternate buffer.

In the third phase of development two new options were added. Option 8 performs a histogram equalization.

This option leaves the original image unchanged while placing the equalized image in the alternate buffer. Histogram equalization maps the image histogram, whose minimum and maximum gray levels are greater than and less than the absolute minimum and maximum gray levels, respectively, into an image with gray levels equally disturbed between all gray levels (0-255 inclusive) [6].

Option 9 makes it easier to use the program. This option copies a digital image from one buffer to the other.

All of the first nine options described were used experimentally to determine the best way to get the required information from the digital images. The last two options were written to extract the needed information.

Option 10 does a double threshold on an image. The user enters an upper and lower threshold. All pixels whose gray level is between the upper and lower limits are left unchanged. All pixels with gray levels outside of the input range are turned black. In this way, areas of bran or endosperm are isolated in the picture. This option always uses the first buffer in memory as the source image and places the resultant image in the second buffer.

Option 11 takes the image in the second buffer and scans it for an area that represents an endosperm or bran particle. This option counts the number of pixels in each area and computes the average gray level of each area. This information is written to an output file, with the filename specified by the user, and with a ".ad" extension. This output file can then be used when desired as an input file for a companion application program called TEST (Appendix A), that compiles and summarizes the input file data.

The TEST program reads in the file with a ".ad" extension and outputs the following information about the image: 1) number of areas isolated; 2) average size of the areas isolated; 3) average gray level of all areas isolated; and 4) percent of total frame covered by the isolated areas. These numbers are the final data that are to be interpreted by the knowledge-based roll gap control system to determine how well the roller mills are performing.

Option 12 terminates the ISOAREA program and returns control to the VAX/VMS operation system.

2.3 Experimental Methods

To demonstrate the utility of the software developed in determining mill stream quality a three part experiment was conducted. The first part took samples milled from pure varieties on the United States Department of Agriculture Grain Marketing Research Laboratory (USDA/GMRL) Buhler experimental mill. (Figure 2.1) The second part took samples milled from a "standard mill mix" on the KSU Grain Science Department's pilot mill. Third, a sample of the KSU mill mix was milled on the USDA/GMRL Buhler experimental mill.

The aim for the first part was to determine if image processing could be used to detect a difference between the features present in pictures of different mill stocks. Specifically, can the software developed detect differences between the three break stocks and the differences between a single break stock milled at three different roll gap settings.

The first decision was to determine what experimental milling method to use. The Buhler experimental mill was chosen because milling results are easier to reproduce on the Buhler mill versus other mills. In addition, the Buhler mill eliminates most of the variables from the milling process.

The second decision was to determine the type of wheat to mill. A pure variety of each hard red winter (HRW) wheat, Newton, and soft red winter (SRW) wheat, Caldwell, was chosen. Pure varieties make control of the

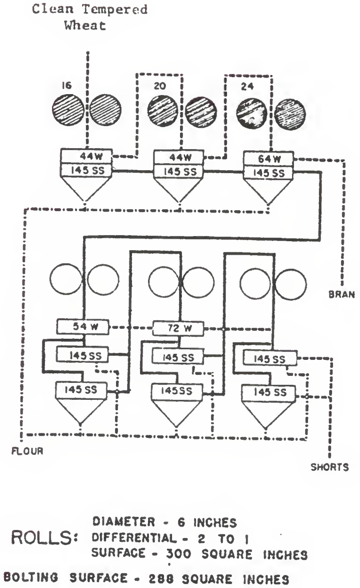


Figure 2.1 Buhler Experimental Mill Flow.

milling experiment easier than mixed wheat blends do. Also, results can be compared to see if the same techniques apply to both classes of wheat.

Since the objective of this study is to control the break system by determining the quality of break streams, the samples of interest are the stock above and below each pair of break rolls. In the Buhler experimental mill all break rolls are mechanically linked so that all the break gaps are adjusted at the same time. Three different gaps settings were selected. (Table 2.1) A sample of each variety was milled at each of the three gap settings and each time a sample of the stock above and below each break roll was taken.

Test	Leftside roll adj.*	Rightside roll adj.*
Newton	17 9 5	17 9 5
Cald-well	17 9 5	17 9 5
Pilot mill blend	17 9 5	17 9 5

*Numbers given are values on Buhler experimental mill roll adjustment handle scale.

Table 2.1 Gaps Used for Buhler Mill Test.

The samples were taken by starting at the point furthest downstream and then working upstream. This technique avoided any lingering effects that taking a sample upstream might have on a sample taken downstream. Care was taken to catch samples across the width of the entire mill stream to preserve the representative sample. Approximately 20 grams of each sample was stored in a sealed plastic bag.

Next, a picture of each sample was taken. The physical layout of the camera, light table, and fluorescent lights is shown in Figure 2.2. The hardest part of this procedure to keep consistent from one sample to another was laying the sample out in front of the camera while preserving the representative nature of the sample. The sample bag was shaken to uniformly mix all particles in the sample. Then, the sample was poured out onto a piece of black colored paper. Next, the sample was carefully flattened by pressing down on the top of the sample with a metal blade (Figure 2.3). This step was necessary to convert the three dimensional pile to an approximate two dimensional plane so the focal point of the camera lens could be ignored. Then, the camera lens was adjusted to bring the sample into the sharpest focus possible. Finally, the picture was taken and stored to

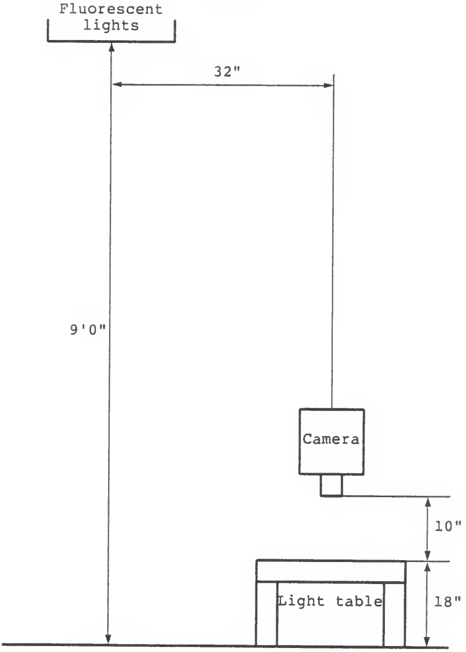
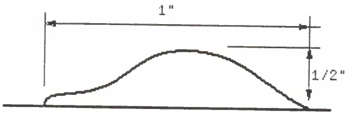
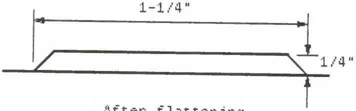


Figure 2.2 Grinnell/VAX Camera and Lighting Layout.



Before flattening



After flattening

All lengths are approximated.

Figure 2.3 Cross Section of Sample as Placed Before Camera.

disk, using the first 5 options of the ISOAREA menu. An example of the digital images obtained appears in Appendix C.

After pictures of all the samples from a single variety were taken, the images were backed up on tape. The next step was to get the bran and endosperm data from the images using the tenth and eleventh options of the ISOAREA menu. First, the picture was loaded into memory from the disk. Next, option 10 was executed with the upper and lower gray levels set so only bran particles remain in the picture. Then, option 11 was executed to calculate the size and average gray value of each area isolated. Option 10 was repeated, this time endosperm particles were isolated. Option 11 was executed a second time to determine the endosperm particle size and average gray value. Finally, the companion program TEST was run using both previously generated ".ad" files to compile and summarize the final data to be analyzed.

The final results are presented using the format shown in Table 2.2. Table 2.2 shows the number of particles isolated, the average size of the isolated particles, the average particle gray level, and the ratio of area covered by the isolated particles to the area of the entire image. Data are presented for bran and endosperm particles separately. Each column represents a

Feature	Roll	Endosperm open	Endosperm stan.	Endosperm close	Bran open	Bran stan.	Bran close
number of particles detected in image	1BK	633	600	618	585	572	568
		422	376	569	712	771	689
	2BK	725	781	--	769	688	--
		485	349	247	561	582	440
	3BK	697	702	--	605	599	--
		408	364	745	--	569	381
average size of particles in pixels	1BK	27	24	22	54	62	65
		35	19	21	45	39	57
	2BK	29	22	--	34	42	--
		8	16	7	85	73	117
	3BK	15	11	--	60	68	--
		20	12	10	--	82	136
average gray level of particles	1BK	175	174	174	124	123	124
		187	185	185	150	147	158
	2BK	173	172	--	126	126	--
		183	184	183	146	145	142
	3BK	172	172	--	125	124	--
		187	185	184	--	145	160
ratio of particle area to image ar.	1BK	26	22	20	48	54	56
		23	11	18	49	46	60
	2BK	32	26	--	40	44	--
		6	8	2	73	65	78
	3BK	15	12	--	56	62	--
		12	6	12	--	71	79

Stock sampled under the rolls.
Top numbers are for Newton(HRW).
Bottom numbers are for Caldwell(SRW).
Boldfaced data is consistent with expected results as roll gap decreases.
Underlined data is consistent with expected results as roll location changes.

Table 2.2 Results of HRW vs. SRW Test on Buhler Mill different roll gap setting, while each row represents the stock under the first break (1BK), second break (2BK), and third break (3BK) rolls, respectively. The top number at each location is the value obtained from the Newton wheat samples while the bottom number at each location is the value obtained from the Caldwell wheat samples.

The missing data in Table 2.2 was inadvertently lost when the images were transferred from the hard disk to the magnetic tape. The data that remains was sufficient to adequately complete the testing of the image processing procedures. Therefore, the time and effort to duplicate the collecting and storage of the missing images was impractical when compared with what additional testing could be accomplished with the additional data.

In the second part of the experiment, "standard mill mix" samples from the KSU Department of Grain Science and Industry pilot mill were analyzed. There were two objectives of this part of the experiment. The first objective was to determine relationships between the changes made in roll gaps and the resultant changes in the features detected in the mill stocks. The second objective will be to encode these relationships as rules in the knowledge-based roll gap control system. This objective will have to be achieved at a later date. Table

2.3 shows the roll gap settings used for the second part of the experiment. At each setting listed, a sample of the mill stock above and below the 1BK, 2BK, and 3BK (coarse) rolls was taken. These samples were analyzed using the same procedure as described for the samples from the Buhler test mill. Table 2.4 shows the data obtained from the analysis of the pilot mill samples. The presentation format is the same as the one used in Table 2.2 for the Buhler experimental mill results.

In the third part of the experiment, a sample from the same "standard mill mix", which was used in the second part of the experiment, was milled on the Buhler experimental mill. The sample collection and handling

	Roll gap settings		
Test	1BK	2BK	3BK
I	30% release	40% release	30% release
II	37% release[1]	previous gap	previous gap
III	25% release[2]	previous gap	previous gap
IV	same as I	46% release[3]	previous gap
V	same as I	30% release[4]	previous gap
VI	same as I	same as I	41% release[5]
VII	same as I	same as I	22% release[6]
VIII	same as II	same as IV	same as VI
IX	same as III	same as V	same as VII

1 Turned roll gap adjustment knob 270° CCW from I.
2 Turned roll gap adjustment knob 360° CW from I.
3 Turned roll gap adjustment knob 540° CW from I.
4 Turned roll gap adjustment knob 540° CCW from I.
5 Turned roll gap adjustment knob 360° CW from I.
6 Turned roll gap adjustment knob 360° CCW from I.

Table 2.3 Gaps Used for Pilot Mill Test.

Feature	Roll	Endosperm open	Endosperm stan.	Endosperm close	Bran open	Bran stan.	Bran close
number of particles detected in image	1BK	315	235	375	649	519	393
	2BK	345	171	219	537	420	579
	3BK	239	246	299	674	409	499
average size of particles in pixels	1BK	21	21	11	57	87	130
	2BK	11	16	13	82	115	78
	3BK	17	9	16	60	116	95
average gray level of particles	1BK	192	191	190	148	143	149
	2BK	191	190	190	146	139	143
	3BK	196	190	191	147	142	148
ratio of particle area to image ar.	1BK	10	7	6	56	68	78
	2BK	6	4	4	67	73	69
	3BK	6	3	7	62	72	72

Stock sampled under the rolls.
Boldfaced data is consistent with expected results as roll gap decreases.
Underlined data is consistent with expected results as roll location changes.

Table 2.4 Results From KSU Pilot Mill Test.

procedures were identical to those used in the first part of the experiment. The image processing procedures used were identical to those used in both of the previous experiment parts. Again, the same presentation format of the results was used. The results appear in Table 2.5.

2.4 Results

The results of the experiment parts, presented in Tables 2.2, 2.4, and 2.5, show data from samples collected from underneath the rolls. The corresponding data for samples collected above the rolls appears in Appendix B. Table 2.2 shows the results from the first part of the experiment, pure varieties on the USDA/GMRL Buhler experimental mill. The results from the second part of the experiment, performed on the KSU Department of Grain Science and Industry pilot mill, are shown in Table 2.4. Table 2.5 shows a comparison of results obtained when the KSU Department of Grain Science and Industry mill mix is milled on both mills.

In general the relationships stated in Table 2.6 hold for the samples of stock collected underneath the rolls. Data in Tables 2.2, 2.4, and 2.5 that are consistent with the relationships related to roll gap are set in boldface type. Data consistent with the

Feature	Roll	Endosperm open	Endosperm stan.	Endosperm close	Bran open	Bran stan.	Bran close
number of particles detected in image	1BK	304 363	235 401	282 491	632 721	519 661	587 749
	2BK	173 590	171 484	208 411	717 654	420 620	596 652
	3BK	295 441	246 468	214 338	538 485	409 674	617 667
average size of particles in pixels	1BK	19 18	21 18	19 16	64 48	87 56	70 50
	2BK	15 18	16 18	17 10	53 66	115 70	71 67
	3BK	14 15	9 9	12 5	81 98	116 64	73 70
average gray level of particles	1BK	197 187	191 187	196 188	150 148	143 150	149 150
	2BK	195 186	190 186	196 184	145 156	139 151	148 149
	3BK	196 185	190 186	194 185	150 153	142 147	147 144
ratio of particle area to image ar.	1BK	9 10	7 11	7 12	62 53	68 57	63 57
	2BK	4 16	4 13	5 6	58 66	73 66	64 66
	3BK	6 10	3 6	4 2	66 73	72 66	68 71

Stock sampled under rolls.
Top numbers are for Pilot Mill.
Bottom numbers are for Buhler Mill.
Boldfaced data is consistent with expected results with respect to changes in roll gap.
Underlined data is consistent with expected results with respect to location of the rolls.

Table 2.5 Buhler/Pilot Mill Comparsion Results.

	Endosperm		Bran	
Feature	wrt gap decrease	wrt break location	wrt gap decrease	wrt break location
number of particles detected in image	increase	decrease	increase	increase
average size of particles in pixels	decrease	decrease	decrease	increase
average gray level of particles	increase	same	same	same
ratio of particle area to image ar.	decrease	decrease	increase	increase

Table 2.6 Expected Qualitative Relationships.

relationships with respect to break location are underlined.

To rationalize the relationships presented in Table 2.6, consider what happens when endosperm and bran particles are ground. As the roll gap decreases, more grinding pressure is placed on the stock. This extra pressure causes an increase in the number of both endosperm and bran particles. The number of endosperm and bran particles is expected to increase as roll gap decreases. Similarly, the average particle size of both bran and endosperm particles is expected to decrease.

Tempering causes mellowing of the endosperm and toughening of the bran suggesting that the endosperm particles decrease in size and increase in number at a faster rate than the bran particles. This means the ratio of endosperm to bran particles is expected to decrease as roll gap decreases.

As stock moves from the 1BK roll to the 3BK roll the percentage of endosperm remaining in the stock decreases while the percentage of bran remaining increases. Expected results are for the endosperm to bran ratio to decrease, the number of endosperm particles to decrease, and the number of bran particles to increase as the stock moves through the break rolls. Corrugated rolls will reduce the size of endosperm particles, while bran particle size tends to increase. It is expected that the average endosperm particle size will decrease and the average bran particle size will increase.

As endosperm particle size decreases its texture becomes finer and the gray level of endosperm particles is expected to increase as roll gap decreases. Initially it was expected that the gray level might be of use to detect the different layers of the wheat berry. While working with the first set of samples it became obvious that the camera was not nearly this sensitive. That is why the value of "same" appears in the remaining columns of the average gray level row of Table 2.6.

Each of the tables containing resultant data can be partitioned into eight separate 9 by 9 grids (the intersection of the four features with the two particle types). Table 2.6 can be partitioned in a like manner. The left half of the Table 2.6 partition contains the expected relationship as data are read across the corresponding data table partition. The right half of the Table 2.6 partition contains the expected relationship as data are read from top to bottom in the corresponding data table partition.

Clearly, all of the data collected is not consistent with the expected results. The inconsistencies are explained by acknowledging the presence of noise in the data and by deficiencies in the feature detection algorithm. Also it is conceivable that a mistake was made in collecting, labeling, and/or obtaining the final data from each sample. The data that is consistent with the relationships of Table 2.6 suggests that with improved data collecting and feature detecting techniques break stream quality can be determined optically.

In general, the measurement of average gray level of the particles provided little information except to distinguish bran particles from endosperm particles. Also, data derived from endosperm particles proved to be more consistent than data derived from bran particles. This suggests that when using the sample collection procedures and image processing procedures of this research, data derived from the endosperm particles are a better predictor of mill stream quality than the data derived from the bran particles. This result seems reasonable for all sample collection procedures and image processing procedures, since the change, caused by the mill break system in bran particles is smaller than the change that occurs in the endosperm particles.

To determine the reliability of the image processing software, the following procedure was performed. The sample handling and image processing procedures described in section 2.3 were repeated five times using the same sample. The sample used was the HRW break stock obtained under the 1BK rolls of the Buhler experimental mill grinding at the standard break release.

The results of each trial are shown in Table 2.7, along with the mean and standard deviation for each feature detected. Results for endosperm and bran particles are shown separately. The last row in each

Endosperm

	number of particles detected in image	average size of particles in pixels	average gray level of particles	ratio of particle area to image ar.
Trial 1	258	23	191	9
Trial 2	342	20	191	10
Trial 3	358	21	192	11
Trial 4	330	22	191	11
Trial 5	290	19	190	8
mean	316	21	191	10
st. dev.	40.9	1.6	0.7	1.3
st. dev. / mean	0.129	0.076	0.004	0.130

Bran

	number of particles detected in image	average size of particles in pixels	average gray level of particles	ratio of particle area to image ar.
Trial 1	679	31	142	32
Trial 2	875	23	143	31
Trial 3	862	26	145	34
Trial 4	824	27	144	34
Trial 5	791	28	142	34
mean	806	27	143	33
st. dev.	78.4	2.9	1.3	1.4
st. dev. / mean	0.097	0.107	0.009	0.042

Table 2.7 Results of Repeatability Test for 1BK Stock.

section of the table is the normalized standard deviation. This was calculated by dividing the original standard deviation by the mean, making the normalized mean equal to one. The largest normalized standard deviation is 0.13, which is for the ratio of endosperm particle area to the total image area. Assuming a normal or Gaussian distribution, the normalized value obtained for the ratio of endosperm particle area to total image area will fall within the narrow range of 1 ± 0.13 68% of the time. These numbers indicate that the procedures used produced reliable readings for each feature detected.

For the HRW vs. SRW test, Table 2.2, the algorithm used worked the best on hard wheat. Since both the HRW and SRW wheats were tempered and milled using the procedures normally used for HRW wheat, this explains the more consistent relationships for the HRW wheat than those obtained for the SRW wheat. Another consideration for eliminating the test result differences between the HRW and SRW wheat is to develop different image processing software for detecting features in the SRW wheat.

Table 2.4 shows results from the tests conducted on the pilot mill. Consistencies can be found in number of particles, average particle size, and ratio of particle area to total frame area. Each of these three parameters is a possible indicator of mill stream quality. Unfortunately the present method of feature detection produces qualitative results only. At this time, no definite relations can be made between the features detected by the image processing system and the features used by the miller.

Table 2.5 shows a comparison between the results obtained from milling the same sample on both the Buhler experimental mill and the Pilot mill. The current algorithm shows little or no difference between the abilities of the two mills to generate samples that produce accurate results.

2.5 Future Work

Since endosperm particles, present in the mill break system, appear to be a better indicator of break stock quality, future efforts should focus on the endosperm particles. Endosperm particles tend to stratify to the bottom of the sample using the described handling procedures, it is recommended that the large bran particles be removed by sifting so the camera can focus in on the endosperm particles.

This procedure is also easier to adapt to an on-line version of the system. Since stratification of smaller particles occurs as the stock exits the rolls, the camera should be placed on the side at which the smaller particles tend to exit. The focal length of the camera lens should be adjusted to maximize the number of endosperm particles detected in the image.

Once the picture is obtained, it is suggested a more sophisticated edge detection algorithm be used to isolate particles in the image. The number of ways to detect edges is numerous and several variations exist for each edge detection algorithm. Two possible ways suggested to the author are gradient detection and contour tracing. Once the particles in the image are isolated, a check of their gray level is used to verify whether or not the particles are indeed endosperm. Other image processing techniques, texture analysis as an example, should be explored as well. Following these suggested procedures are expected to improve the reliability of the results. Data derived from information contained in the endosperm particles can be used to control the roller mills.

Chapter 3

3.0 Development of a Prototype Roll Gap Control System

This chapter of the thesis contains a description of how the data obtained using the image analysis system can be utilized by an expert system to control the break roll gap. Presently it is planned to develop and implement a prototype roll gap knowledge-based control system using an expert system shell.

3.1 Knowledge Required for Roll Gap Control System

Several types of knowledge are required for the roll gap knowledge-based system to function. The flow of the break system is one type of knowledge required. For initial prototype development it is desired to start with a simple flow. This discussion centers on a flow of stock through the first three breaks of a five break mill. A simple flow diagram for the system is shown in Fig. 3.1.

This flow was chosen for three reasons. First, the first three breaks have the biggest impact on mill balance and the resultant extraction rates. Improving the control of the first three breaks will improve mill performance the most. Second, knowledge-based systems are developed incrementally. A prototype is built that works for a small portion of the original problem domain and then is gradually expanded to cover a larger portion of the

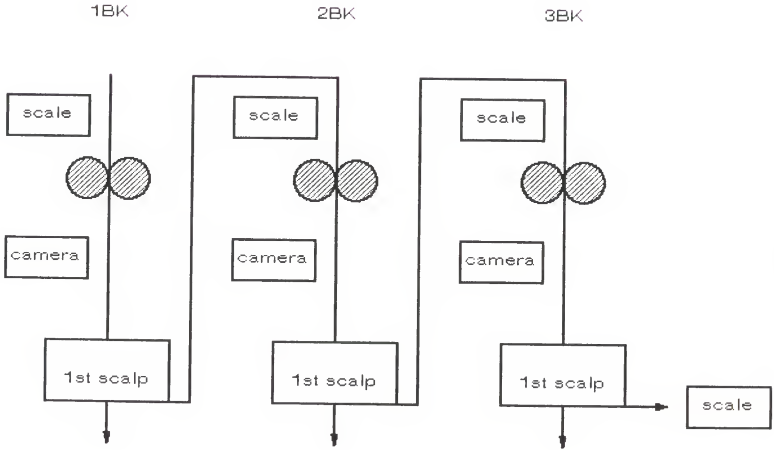


Figure 3.1 Flow Used for Prototype Roll Gap Control System.

problem domain. The first three breaks make a good first increment for the roll gap knowledge-based system. Third, the time necessary to construct a large system is much greater than is available to complete this thesis.

Another type of knowledge required by the roll gap control system is the objective the system is to achieve. In this case the objective for the first three breaks is to open up the wheat berry, break off the bulk of the endosperm and suitably size it in preparation for further processing [7]. Another objective is to minimize the amount of bran powder produced by each pair of rolls. Since these objectives can not be fully met simultaneously, the miller optimizes the breaking process by closing the rolls until the extra benefit gained by the extra release of endosperm is overcome by the cost of increased bran powder production.

3.2 Knowledge Used to Adjust Roll Gap

A third type of knowledge is the heuristics used by the miller to adjust the roll gaps. The miller starts with the first break roll and then moves to the second break roll followed by the third break roll. Adjustments are made in this order because the roll gap setting of one roll affects the quality of the stock flowing into the next roll.

First, the miller checks the flow of stock into the nip of the rolls. The feed gate is adjusted, if necessary, to ensure even feeding of stock along the entire grinding surface. Second, the miller checks for the evenness of grind. This is done by running the hands underneath the roll and feeling the warmth of the stock as it leaves the rolls. If the stock at one end of the rolls is warmer than at the other end, the rolls are grinding unevenly. Visual checks of the stock as it leaves the rolls can be made as well.

Third, the miller uses a test sifter to determine the break release of the rolls. The break release can be estimated by the miller using visual inspection, but test sifting is the only accurate way to set the roll gap to a specified break release. Finally, the miller fine tunes the break roll gap setting. This usually occurs after the miller has received some feedback, from the lab, regarding the quality of the flour being produced.

3.3 Formal Representation of the Knowledge

Now that the knowledge required to adjust the roll gap of the first three breaks has been identified, it can be formalized. This step requires organization of the knowledge into a data structure. In turn, the data structure can be easily represented and manipulated in the memory of a computer.

Figure 3.2 shows a highly simplified version of a goal tree used by the miller when setting the first three break roll gaps. The top level goal is to set all three breaks. This goal is broken down into three sub-goals, one for each break. In this case, order is important. The goals at the same level are solved from left to right. Each sub-goal is further subdivided into three more sub-goals. The first of these is to ensure proper flow of the stock into the nip of the rolls. The next goal is to ensure the rolls are aligned in parallel. The final sub-goal is to adjust the roll gap to the desired break release (Figure 3.2).

Now the problem becomes, "How can the knowledge-based system achieve the same goals as the miller?" Obviously the outputs of the knowledge-based control system are the desired break roll gaps. What remains is to define the inputs to the knowledge-based system, what information each input provides, and how the information is used to make the same decisions made by the miller.

Figure 3.1 shows the proposed locations of the seven instruments that will collect all necessary input information. Underneath each pair of rolls a digitizing camera is placed to capture images of the stock as it

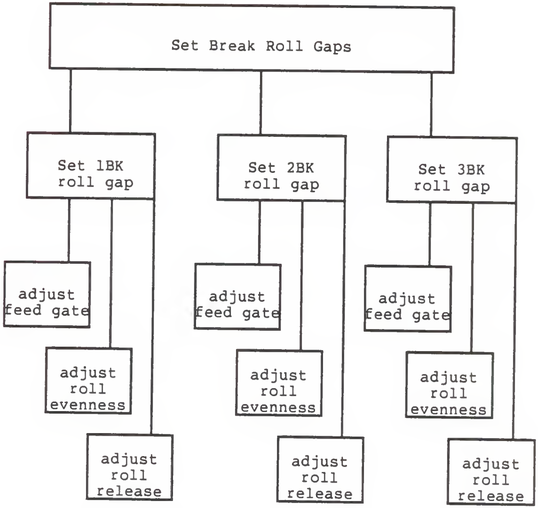


Figure 3.2 Simplified Miller's Goal Tree.

exits the roll. In-line scales are placed above each roll and on the stock going to the 4th break to measure the amount of stock flowing into each break.

The in-line scales provide break release information, by simply subtracting the values from two adjacent scales in the flow. This information is used to maintain the balance of the mill load. It is the same type of information obtained by the miller when test sifting is used to initially set the roll gaps.

The three digitizing cameras provide images of the stock as it leaves each break. The images are analyzed using the techniques outlined in Chapter 2. This information is used to determine the desired roll gap. It is analogous to the information the miller uses to fine tune the break system. The difference is the source of the information. The miller usually fine tunes the break roll gap settings from feed back on the quality of the flour being produced, while the knowledge-based system fine tunes the break roll gap settings based on break stock quality.

At this time, no automatic adjustment of the feed gate is proposed. Nor is any information about the flow of the stock into the nip of the rolls automatically acquired. The knowledge-based system will have to either assume the feed gates are adjusted properly 100% of the time or receive an input signal from the miller immediately after the feed gates have been adjusted and then assume they remain adjusted for a certain period of time. If the latter case is chosen, the knowledge-based system can be programmed to periodically remind the miller to check the feed gate adjustments.

Assuming the roll gap controlling mechanism is manufactured at a high level of precision, the rolls should always be operating in a parallel alignment. Referring back to Fig. 3.2, the first two low level goals have been satisfied at this point. The knowledge-based control system needs to concern itself with the third low level goal. To accomplish this task a new goal tree, Fig. 3.3, has been developed. This goal tree can easily be implemented in any knowledge-based shell that has a backward chaining inference engine. How a backward chaining inference engine makes a decision will be described shortly. First, for demonstration purposes, some assumptions about the input information received by the knowledge-based system are made.

Assume the recommended steps in Chapter 2 are carried out. The results show that endosperm particle size, in the mill break system is the best indicator of stock quality. Additionally, assume further experimentation

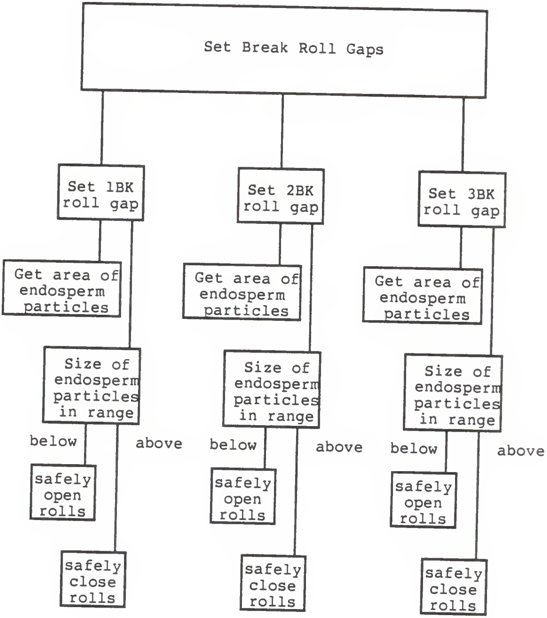


Figure 3.3 Goal Tree for Roll Gap Control System.

determines optimal ranges for the endosperm particle size at each break. Now simple IF-THEN rules can be written to make the proper decision of what the break roll gap should be. An example rule set is listed in Table 3.1. Notice how the rules are linked together by matching clauses between one rule's IF clause and another rule's THEN clause. These links are traced by a backward chaining inference engine to satisfy the top level goal. To demonstrate how the knowledge-based system will work, a description of the execution of a backward chaining inference engine using the rules in Table 3.1 follows.

3.4 Example Reasoning Strategy

The backward chaining inference engine starts by trying to satisfy the top level goal. In this case the top level goal is represented by the THEN clause of rule 1. To satisfy the top level goal, the inference engine finds that it must satisfy three sub-goals, the three IF clauses of rule 1. The inference engine satisfies sub-goals one at a time, starting with the first clause. In this case, the first sub goal is "1BK roll gap is set".

Rules 2, 3, and 4 conclude the first sub-goal. If the inference engine can prove that all of the premises of any one of these three rules, is true it will have satisfied the first sub-goal. Rule 2 premises, IF

Rule 1		
	IF:	1BK roll gap is set and 2BK roll gap is set and 3BK roll gap is set
	THEN:	set break roll gaps goal is satisfied
Rule 2		
	IF:	average size of endosperm particles below the first break roll is known and average size is within the predetermined optimum range
	THEN:	1BK roll gap is set and output no change in 1BK roll gap made
Rule 3		
	IF:	average size of endosperm particles below the first break roll is known and average size is above the predetermined optimum range and 1BK rolls can be closed
	THEN:	1BK roll gap is set and output close 1BK roll gap by one increment
Rule 4		
	IF:	average size of endosperm particles below the first break roll is known and average size is below the predetermined optimum range and 1BK rolls can be opened
	THEN:	1BK roll gap is set and output open 1BK roll gap by one increment
Rule 5		
	IF:	the procedure for obtaining the average size of the endosperm particles below the first break roll has been called and the procedure has been successfully completed
	THEN:	average size of endosperm particles below the first break roll is known and store the returned value for future reference
Rule 6		
	IF:	the weight of the stock released by the first break roll does not overload downstream equipment and the rolls will not touch if closed further
	THEN:	1BK rolls can be closed
Rule 7		
	IF:	the weight of the stock released by the first break roll does not underload downstream equipment
	THEN:	1BK rolls can be opened

Table 3.1. If-Then Rules for Knowledge-Based System.

clauses, are tried first. The first IF clause asks the question, "Is the average endosperm particle size known?" To answer this question the inference engine backward chains to rule 5.

The execution of rule 5's premises causes the knowledge-based system to obtain an image of the first break stock and analyze it for the desired quantity. After the image has been analyzed, rule 5 THEN clause is executed, satisfying the first sub-goal of rule 2. In addition, a side effect occurs when rule 5 THEN clause is executed. This side effect stores the obtained average endosperm particle size in memory for future reference.

Now the backward chaining inference engine is ready to test the second IF clause of rule 2. This clause tests the value returned by rule 5. If the first break roll gap is already set in the optimum range, the second clause of rule 2 is true and the THEN clause of rule 2 is executed. The execution of the THEN clause in rule 2 satisfies the first sub-goal of rule 1 and creates the side effect of generating an output of no change to the roll gap control mechanism on the first break roll.

In the case where the 1BK roll gap is too wide, the second premise of rule 2 would fail. At this point the inference engine will continue to rule 3. The first premise of rule 3 has already been traced. The inference engine moves directly to the second premise. Again, assuming the first break roll gap is too wide, this premise will be true.

Now, the inference engine moves to the third premise of rule 3. To satisfy this sub-goal, the backward chaining inference engine moves to rule 6 to see if the 1BK rolls can be safely closed. If rule 6 fires, meaning all premises of rule 6 are true, the THEN clause is executed. The third clause of rule 3 is true and its THEN clause is executed. This action causes the first sub-goal of rule one to be satisfied and the side effect of closing the first break roll gap by one increment is executed as well. Similar reasoning occurs when the first break roll gap is too narrow, except in this case rule 4 is used to adjust the break gap.

Once the first sub-goal of rule one is satisfied, the backward chaining inference engine will move on the second sub-goal and adjust the second break roll gap in a similar manner. Finally, it will adjust the third break roll gap. At this point all three sub-goals have been satisfied and rule 1 THEN clause is executed, which satisfies the top level goal. The backward chainer ceases it search.

For continuous operation, the top level goal is reinstantiated. This action causes all facts determined during the previous instantiation to be forgotten and the adjustment procedure starts over again.

The control strategy can be modified several ways. For example, the first sub-goal could be continuously reinstantiated until rule 2 fires. Setting the first break roll gap at its optimum setting before moving to the second sub-goal. Also, the thresholds that are used to compare the average endosperm particle size to determine whether or not the rolls can be safely opened or closed could be modified, dependent on certain exterior conditions, e.g., wheat mix, flour specifications, etc. This would require an additional initialization sub-goal added to the beginning of rule 1. To satisfy this goal the exterior conditions would be checked and the appropriate threshold values selected. Quickly, this simple demonstration control strategy becomes quite complicated.

Other complications will arise during the implementation of even the simplest control strategy. For example, when determining the loading threshold of the downstream equipment the knowledge-based system developer must keep in mind that most of the equipment receives its load from more than one source. A situation could arise where it was determined that both the first and second break rolls needed to be closed, but a purifier that both breaks feed could not handle the combined extra load.

At this point it is clear why the original scope of the problem was limited. Development of such a system becomes complicated very rapidly. With the use of an expert system shell software package running on a standard personal computer, the prototype system described could be developed over a period of a few months.

Chapter 4

4.0 Conclusions

The primary goal of this research was to determine whether or not image processing technology can be used to detect break stream quality. Although, the data collected was not completely consistent with expected results, the data suggested, with improved image processing techniques and sample handling procedures, that break stream quality can be determined optically.

The secondary goal was to find out which image processing technique or techniques work the best. Because of the lack of work on this goal, no conclusions can be made. However, the reliability of the image processing techniques used was determined to have a normalized standard deviation of ± 0.13.

The experiment was conducted in three parts. In the first part pure varieties of HRW and SRW were milled on the Buhler experimental mill. For this portion of the experiment, the image processing techniques used, showed that SRW wheat must be tempered and milled differently than the HRW wheat. The tempering, milling, and image processing software was biased toward the HRW wheat.

The second part of the sample collection experiment involved milling a standard hard wheat mill mix on the pilot mill and independently adjusting the gap of the 1BK, 2BK, and 3BK (coarse) rolls. The results of this part showed that the features detected from the endosperm particles indicate the quality of the mill break streams better than the bran particles. This result was expected since the change that occurs to the endosperm particles within the mill break system is greater than the change that occurs to the bran particles.

For the third part of the sample collection experiment, the same standard hard wheat mill mix was used as that used in the second part. The mill mix was milled on both the pilot mill and the Buhler experimental mill. Only minor differences were noted between the results obtained from each mill.

Finally, a discussion was presented on how the results could be used to develop a knowledge based control system that controls the roll gaps. Once the relationships between the features detected in the break stock and the optimal break release are finalized, a prototype roll gap control system can be constructed using the knowledge representation discussed.

References

[1] Wingfield, John. "Flour Mill Control." Association of Operative Millers Technical Bulletins, September 1974, pp. 3477-3481.

[2] Clegg, Richard A. "A closer Look at Automated Milling Technology." Association of Operative Millers Technical Bulletins, October 1988, pp. 5303-5308.

[3] Posner, Elieser S., and David L. Wetzel. "Control of Flour Mills by NIR On-Line Monitoring." Association of Operative Millers Technical Bulletins, April 1986, pp. 4711-4720.

[4] Fowler, Eddie, R. "Applications of Artificial Intelligence in Mill Process Control." Association of Operative Millers Technical Bulletins, forthcoming.

[5] Devore, John. "Image Library Software." Kansas State University. March 1983.

[6] Rosenfeld, Azriel, and Avinosh Kac. Digital Picture Processing. 2nd. ed. New York, N.Y.: Academic Press, 1982. vol. 1, pp. 167-174.

[7] Association of Operative Millers. Correspondence Course in Flour Milling. Kansas City, Missouri, 1950.

Appendix A. ISOAREA Source Code

```
C***********************************************************
c
c       Main-routine
c               ISOAREA
c
c       Filename
c               ISOAREA.FOR
c
c       Execution sequence
c               run isoarea
c
c       Description
c               Isoarea is the main program.  It is a
c               single menu driven program with twelve
c               options.  The program prompts the user
c               for the option to execute, calls the
c               appropriate subroutine, and redisplays
c               the menu on the screen.
c
c       Routine(s) called by this routine
c               IMGETS     user written routine
c               IMSHOW     user written routine
c               IMSAVE     user written routine
c               IMLOAD     user written routine
c               HIST       user written routine
c               THRS       user written routine
c               EQUAL      user written routine
c               ENDO       user written routine
c               AREA       user written routine
c
c               IMINIT     image library routine
c               GRDDG      Grinnell library routine
c               GRSBFD     Grinnell library routine
c               GRSEND     Grinnell library routine
c
c       Routine(s) called by embedded routines
c               IMOPEN     image library routine
c               IMTRAN     image library routine
c               IMDISP     image library routine
c
c       Author
c               Keith W. Hefty
c
c       Date
c               22 June 1988
c
c       Revisions
```

```
c               20 July 1988  Program version 4
c                             renamed PROCIMAG to ISOAREA
c                             changed to 256 by 256 images
c               5 August 1988  added option 10
c               15 September 1988  added option 11
c*****************************************************
c
      integer adata(256,256),bdata(256,256),ixloc,iyloc,
     & unit,row,col,opt,hi,lo,hgram(256),buffer,
     & threshold,eq,upt,lot
      character*20 type,prompt,letter,filename
c
c     Initialize program variables.
c
      ixloc=128
      iyloc=128
      unit=1
      type='INTEGER'
      prompt='Enter image filename '
c
c     Initialize Grinnell hardware.
c
      call iminit ('nil')
c
c     Print menu on screen.
c
20    print *,1,'  Get picture'
      print *,2,'  Display picture'
      print *,3,'  Save picture'
      print *,4,'  Load picture'
      print *,5,'  Take picture'
      print *,6,'  Histogram image'
      print *,7,'  Threshold image'
      print *,8,'  Equalize image'
      print *,9,'  Transfer data between buffers'
      print *,10,'  Isolate gray level range'
      print *,11,'  Collect area data'
      print *,12,'  Exit'
      print *,'    Enter number of operation to execute'
      accept *,opt
c
c     Execute menu selection.
c
c     Option 1 gets an image from Grinnell and places
c     it in the VAX's main memory.
c
      if(opt.eq.1)then
         print *,1,' a-buffer'
         print *,2,' b-buffer'
```

```
          print *,'  Enter buffer number'
          accept *,buffer
          if(buffer.eq.1)then
             call imgets (ixloc,iyloc,type,adata)
          end if
          if(buffer.eq.2)then
             call imgets (ixloc,iyloc,type,bdata)
          end if
       end if
c
c      Option 2 displays an image in the VAX memory on
c      the Grinnell monitor.
c
       if(opt.eq.2)then
          print *,'Enter the lower left-hand corner'
          print *,'coordinates of the image'
          accept *,ixloc,iyloc
          print *,1,' a-buffer'
          print *,2,' b-buffer'
          print *,'   Enter buffer number'
          accept *,buffer
          if(buffer.eq.1)then
             call imshow (ixloc,iyloc,type,adata)
          end if
          if(buffer.eq.2)then
             call imshow (ixloc,iyloc,type,bdata)
          end if
          ixloc=128
          iyloc=128
       end if
c
c      Option 3 saves an image in memory to disk.  The
c      user is prompted for a filename from the
c      image library subroutine, IMOPEN.  A ".dat"
c      extension is assumed.
c
       if (opt.eq.3)then
          print *,1,' a-buffer'
          print *,2,' b-buffer'
          print *,'  Enter buffer number'
          accept *,buffer
          if(buffer.eq.1)then
             call imsave (type,unit,prompt,adata)
          end if
          if(buffer.eq.2)then
             call imsave (type,unit,prompt,bdata)
          end if
       end if
c
```

```
c       Option 4 loads a picture into memory from the
c       disk.  The user is prompted for a filename from
c       the image library subroutine, IMOPEN.  A ".dat"
c       extension is assumed.
c
        if(opt.eq.4)then
           print *,1,' a-buffer'
           print *,2,' b-buffer'
           print *,'  Enter buffer number'
           accept *,buffer
           if(buffer.eq.1)then
              call imload (type,unit,prompt,adata)
           end if
           if(buffer.eq.2)then
              call imload (type,unit,prompt,bdata)
           end if
        end if
c
c       Option 5 operates the Grinnell digitizer.  The
c       first call to grddg turns the digitizer on.  The
c       second call turns the digitizer off.
c
        if(opt.eq.5)then
           call grddg (1,2,1,7)
           call grsbfd
           print *,'Press return to take picture'
           accept 100,letter
           call grddg (1,0,1,7)
           call grsbfd
        end if
100     format(a)
c
c       Option 6 perfoms a histogram on an image
c       in memory.
c
        if(opt.eq.6)then
           print *,1,' a-buffer'
           print *,2,' b-buffer'
           print *,'  Enter buffer number'
           accept *,buffer
           if(buffer.eq.1)call hist (adata,hgram,hi,lo,1)
           if(buffer.eq.2)call hist (bdata,hgram,hi,lo,1)
        end if
c
c       Option 7 thresholds a image in memory.
c
        if(opt.eq.7)then
           print *,1,' a-buffer into b-buffer'
           print *,2,' b-buffer into a-buffer'
```

```
            print *,'  Enter buffer number'
            accept *,buffer
            print *,'Enter threshold (0 255)'
            accept *,threshold
            if(buffer.eq.1)then
               call thrs (adata,bdata,threshold)
            end if
            if(buffer.eq.2)then
               call thrs (bdata,adata,threshold)
            end if
         end if
c
c        Option 8 equalizes an image's histogram.
c
         if(opt.eq.8)then
            print *,1,' a-buffer into b-buffer'
            print *,2,' b-buffer into a-buffer'
            print *,'  Enter buffer number'
            accept *,buffer
            print *,'Options'
            print *,1,' Stretch Histogram'
            print *,2,' Equalize Historgam'
            accept *,eq
            if(buffer.eq.1)then
               call hist (adata,hgram,hi,lo,0)
               call equal (adata,bdata,hgram,hi,lo,eq)
            end if
            if(buffer.eq.2)then
               call hist (bdata,hgram,hi,lo,0)
               call equal (bdata,adata,hgram,hi,lo,eq)
            end if
         end if
c
c        Option 9 transfers an image between the two image
c        arrays in the VAX's memory.
c
         if(opt.eq.9)then
            print *,1,' a-buffer into b-buffer'
            print *,2,' b-buffer into a-buffer'
            print *,'  Enter buffer number'
            accept *,buffer
            if(buffer.eq.1)then
               do row=1,256
                  do col=1,256
                     bdata(row,col)=adata(row,col)
                  end do
               end do
            end if
            if(buffer.eq.2)then
```

```
                do row=1,256
                   do col=1,256
                      adata(row,col)=bdata(row,col)
                   end do
                end do
             end if
          end if
c
c         Option 10 isolates bran or endosperm particles
c         based upon individual pixel gray levels.
c
          if(opt.eq.10)then
             print *,'Enter the lower and upper'
             print *,'threshold levels'
             accept *,lot,upt
             call endo (adata,bdata,upt,lot)
          end if
c
c         Option 11 calculates the data for the isolated
c         particles.
c
          if(opt.eq.11)then
             print *,'Enter filename to put area data in:   '
             accept 500,filename
             call area(bdata,filename)
          end if
500       format(a)
c
c         Option 12 terminates the program.
c
          if(opt.eq.12)then
             call grsend
          else
             goto 20
          end if
c
          end
```

```
C**********************************************************
c
c       Subroutine
c               IMGETS
c
c       Filename
c               IMGETS.FOR
c
c       Calling sequence
c               call  imgets (ixloc,iyloc,type,array)
c
c       Description
c               Imgets gets a 256 by 256 image from the
c               center of the Grinnell Screen.
c
c       Argument(s) required from the calling routine
c               ixloc    integer
c                        The x coordinate for the lower
c                        left hand corner of the image.
c                        (0 511)
c                        The default is 128, which allows
c                        for a 256 by 256 picture to be
c                        grabbed from the center of the
c                        512 by 512 Grinnell screen.
c
c               iyloc    integer
c                        The y coordinate for the lower
c                        left hand corner of the image.
c                        (0 511)
c                        The default is 128, which allows
c                        for a 256 by 256 picture to be
c                        grabbed from the center of the
c                        512 by 512 Grinnell screen.
c
c               type     character*20
c                        The format of the data that is
c                        returned.  The default is
c                         'INTEGER'.
c
c       Argument(s) supplied to the calling routine
c               array    integer(256,256)
c                        Image from the center of the
c                        Grinnell screen.
c
c       Routine(s) called by this routine
c               IMDISP   image library function
c
c       Author
c               Keith W. Hefty
```

```
c
c       Date
c               4 March 1988
c
c       Revisions
c               20 July 1988  changed to 256 by 256 images
c
c*******************************************************
c
        subroutine imgets (ixloc,iyloc,type,array)
c
c       gets a 256 by 256 image from center of Grinnell
c       screen
c
        integer array(256,256),nelem,melem,nline,
c    &  ixloc,iyloc
        character*20 type
c
c       Initialize variables to pass to IMDISP routine.
c
        melem=256
        nelem=256
        nline=256
c
c       Get image from Grinnell.
c
        call imdisp('READ',type,array,melem,nelem,nline,
     &  ixloc,iyloc,'WHITE')
c
        return
        end
```

```
C*****************************************************************
C
C       Subroutine
C               IMSHOW
C
C       Filename
C               IMSHOW.FOR
C
C       Calling sequence
C               call imshow (ixloc,iyloc,type,array)
C
C       Description
C               Imshow displays a 256 by 256 pixel image
C               in VAX memory on the Grinnell screen.
C               The user is prompted for the lower left
C               hand corner of the image by the calling
C               routine.  Four images can be displayed at
C               once by choosing the coordinates:
C                (0, 0), (0, 256), (256, 0), (256, 256).
C               An image can be displayed at the center
C               of the screen by choosing the coordinates
C               (128,  128).
C
C       Argument(s) required from the calling routine
C               ixloc    integer
C                        The x coordinate for the lower
C                        left hand corner of the image.
C
C               iyloc    integer
C                        The y coordinate for the lower
C                        left hand corner of the image.
C
C               type     character*20
C                        The format of the image data
C                        being passed in.  Default is
C                         'INTEGER'.
C
C               array     integer(256,256)
C                        The image data.
C
C       Argument(s) supplied to the calling routine
C               None.
C
C       Routine(s) called by this routine
C               IMDISP    image library routine
C
C       Author
C               Keith W. Hefty
C
```

```
c       Date
c               4 March 1988
c
c       Revisions
c               20 July 1988  changed to 256 by 256 images
c
c**********************************************************
c
        subroutine imshow (ixloc,iyloc,type,array)
c
c       Displays a 256 by 256 image on the Grinnell
c       screen.  Data displayed is passed in using array.
c
        integer array(256,256),melem,nelem,nline,ixloc,iyloc
        character*20 type
c
c       Initialize variables to pass to image library
c       routine.
c
        melem=256
        nelem=256
        nline=256
c
c       Write data to Grinnell screen.
c
        call imdisp ('WRITE',type,array,melem,nelem,nline,
     &  ixloc,iyloc,'WHITE')
c
        return
        end
```

```
c*********************************************************
c
c       Subroutine
c               IMSAVE
c
c       Filename
c               IMSAVE.FOR
c
c       Calling sequence
c               call imsave (type,unit,prompt,array)
c
c       Description
c               Imsave stores a 256 by 256 pixel image to
c               a disk file.  The user is prompted for a
c               filename from the image library routine,
c               IMOPEN.  An extension of ".dat"
c               is assumed.
c
c       Argument(s) required from the calling routine
c               type      character*20
c                         A string that specifies the
c                         format of the data being passed
c                         to the image library routine.
c                         The default is 'INTEGER'.
c
c               unit      integer
c                         Logical unit specifier (0 99).
c
c               prompt    character*20
c                         A string used by the image
c                         library routine, IMOPEN, to
c                         prompt the user for a filename.
c
c               array     integer(256,256)
c                         The image data to be stored.
c
c       Argument(s) supplied to the calling routine
c               None.
c
c       Routine(s) called by this routine
c               IMOPEN    image library routine
c               IMTRAN    image library routine
c
c       Author
c               Keith W. Hefty
c
c       Date
c               4 March 1988
c
```

```
c       Revisions
c               20 July 1988  changed to 256 by 256 images
c
c**********************************************************
c
        subroutine imsave (type,unit,prompt,array)
c
c       Stores a 256 by 256 image in a disk file.
c
        integer array(256,256),melem,nelem,nline,unit
        character*20 type,prompt
c
c       Initialize variables to pass to image library
c       routine.
c
        melem=256
        nelem=256
        nline=256
c
c       Open the file.
c
        call imopen (unit,'WRITE',prompt,'NONAME',nelem,nline)
c
c       Write image data to the file.
c
        call imtran (unit,'WRITE',type,array,melem,nelem,nline)
c
c       Close the file and return.
c
        close(unit)
        return
        end
```

```
C************************************************************
c
c       Subroutine
c               IMLOAD
c
c       Filename
c               IMLOAD.FOR
c
c       Calling sequence
c               call imload (type,unit,prompt,array)
c
c       Description
c               Imload loads a 256 by 256 pixel image
c               from a disk file.  The user is prompted
c               for the file name from the image library
c               subroutine, IMOPEN.  An extension of
c               ".dat" is assumed.
c
c       Argument(s) required from the calling routine
c               type      character*20
c                         A string that specifies the
c                         format of the image data
c                         returned.  The default value is
c                          'INTEGER'.
c
c               unit      integer
c                         Logical unit specifier (0 99).
c
c               prompt    character*20
c                         A string used by the image
c                         library routine, IMOPEN, to
c                         prompt the user for a filename.
c
c       Argument(s) supplied to the calling routine
c               array     integer(256,256)
c                         The image data loaded.
c
c       Routine(s) called by this routine
c               IMOPEN    image library routine
c               IMTRAN    image library routine
c
c       Author
c               Keith W. Hefty
c
c       Date
c               4 March 1988
c
c       Revisions
c               20 July 1988  changed to 256 by 256 images
```

```
c
c*********************************************************
c
      subroutine imload (type,unit,prompt,array)
c
c     Loads a 256 by 256 image from a disk file and
c     puts it in an array.
c
      integer array(256,256),nelem,melem,nline,unit
      character*20 type,prompt
c
c     Initialize variables to pass to image library
c      functions.
c
      melem=256
      nelem=256
      nline=256
c
c     Open image file.
c
      call imopen (unit,'READ',prompt,'NONAME',nelem,nline)
c
c     Read data from file.
c
      call imtran (unit,'READ',type,array,melem,nelem,nline)
c
c     Close file and return.
c
      close(unit)
      return
      end
```

```
c*********************************************************
c
c       Subroutine
c               HIST
c
c       Filename
c               HIST.FOR
c
c       Calling sequence
c                call hist (array,hgram,hi,lo,su)
c
c       Description
c               Hist computes the histogram of gray levels
c               present in a 256 by 256 pixel image.  For
c               display on the terminal screen, the number
c               of gray levels is reduced to 64 and the
c               histogram is normalized to a maximum value
c               of 22.  The user has the option of saving
c               the screen histogram to a file for output
c               to a printer at a later time.
c
c       Argument(s) required from the calling routine
c               array    integer(256,256)
c                        The image data in integer format.
c
c               su       integer
c                        Su (screen update) has a value
c                        of 1 or 0.  When a 1 is passed
c                        the subroutine prints the
c                        historgram to the terminal
c                        screen.  If a zero is passed
c                        output to the screen is disabled.
c
c       Argument(s) supplied to the calling routine
c               hgram    integer(256)
c                        This array contains the number of
c                        occurences of each gray level
c                        in the image.
c
c               hi       integer
c                        Value of the highest gray level
c                        detected in the image.
c
c               lo       integer
c                        Value of the lowest gray level
c                        detected in the image.
c
c       Routine(s) called by this routine
c               None.
```

```
c
c       Author
c               Keith W. Hefty
c
c       Date
c               5 March 1988
c
c       Revisions
c               24 July 1988  updated to 256 by 256 images
c
c*************************************************************
c
        subroutine hist (array,hgram,hi,lo,su)
c
c       Performs histogram on a image stored as integer
c       data in a 256 by 256 array.
c
        integer hgram(256),rdata(64),array(256,256),i,j,
     &  row,col,max,pt,su
        character*1 p(23),letter,plot(23,64)
        character*20 fname
c
c       Initialize histogram array to zero.
c
        do i = 1,256
           hgram(i)=0
        end do
c
c       Compute histogram and find high and low
c       gray levels.
c
        hi=1
        lo=256
        do row = 1,256
           do col = 1,256
              i=array(row,col) + 1
              hgram(i)=hgram(i) + 1
              if(array(row,col).gt.hi)hi=array(row,col)
              if(array(row,col).lt.lo)lo=array(row,col)
           end do
        end do
c
c       Print histogram.
c
c       Reduce data to 64 gray levels.
c
        do i=1,64
           j=4*i
           rdata(i)=hgram(j-3)+hgram(j-2)+hgram(j-1)+
```

```
     &     hgram(j)
        end do
c
        max=0
        do i=1,64
           rdata(i)=rdata(i)/4
           if(rdata(i).gt.max)max=rdata(i)
        end do
c
c       Normalize data to maximum value of 22.
c
        do i=1,64
           rdata(i)=22*rdata(i)/max
        end do
c
c       Convert data to screen picture.
c
        do j=1,64
           do i=1,23
              plot(i,j)=' '
           end do
           pt=rdata(j)+1
           plot(24-pt,j)='X'
           do i=25-pt,23
              plot(i,j)='.'
           end do
        end do
c
c       Display histogram on screen.
c
        if(su.eq.0) goto 130
        do i=1,23
           print 100,(plot(i,j),j=1,64)
        end do
c
c       Prompt user for name of file to write histogram.
c
        print *,'Enter filename to save plot'
        print *,'Enter N to exit without saving'
        accept 90,fname
90      format (a)
c
c       Write character map to file.  When an 'N' is
c       entered as a filename this routine return
c       immediately and no file is written.
c
        if(fname.eq.'N'.or.fname.eq.'n')goto 130
c
        open(unit=5,file=fname,defaultfile='.PIC',
```

```
     &   status='NEW')
         do i=1,23
            write(5,100) (plot(i,j),j=1,64)
         end do
100      format(' ',64a)
         write(5,110) 'histogram for data in file ',fname,
     &   'max = ',max
110      format(' ',a27,a20,a6,i5)
         close(5)
c
130      return
         end
```

```
C**********************************************************
c
c       Subroutine
c               THRS
c
c       Filename
c               THRS.FOR
c
c       Calling sequence
c               call thrs (array,tdata,threshold)
c
c       Description
c               Thrs performs a threshold on 256 by 256
c               image in memory.  To display the results
c               use IMSHOW with the threshold data.
c
c       Argument(s) required from the calling routine
c               array           integer(256,256)
c                               The image data, in integer
c                               format, to be threshold.
c
c               threshold       integer
c                               Gray level threshold.  All
c                               pixels whose gray level is
c                               below the threshold are set
c                               to a gray level of 0 (black).
c                               All other pixels have their
c                               gray level set to 255
c                               (white).
c
c       Argument(s) supplied to the calling routine
c               tdata           integer(256,256)
c                               The threshold image.
c
c       Routine(s) called by this routine
c               None.
c
c       Author
c               Keith W. Hefty
c
c       Date
c               10 June 1988
c
c       Revisions
c               20 July 1988  changed to 256 by 256 images
c
C**********************************************************
c
        subroutine thrs (array,tdata,threshold)
```

```
c
c       Thresholds a 256 by 256 pixel image.
c
        integer row,col,array(256,256),threshold,
     &  tdata(256,256)
c
c       Threshold data in 'array'.
c       Store result in 'tdata'.
c
        do row=1,256
           do col=1,256
              if(array(row,col).lt.threshold)then
                 tdata(row,col)=0
              else
                 tdata(row,col)=255
              end if
           end do
        end do
c
        return
        end
```

```
c*****************************************************************
c
c       Subroutine
c               EQUAL
c
c       Filename
c               EQUAL.FOR
c
c       Calling sequence
c               call equal (idata,edata,higram,hi,lo,ty)
c
c       Description
c               The equal subroutine has two options.
c               The first option stretches the histogram
c               so that it covers the whole range of
c               possible gray levels.  The second option
c               levels out the peaks in orginal histrogram
c               by trying to create an even distrubution
c               of gray levels.
c
c       Argument(s) required from the calling routine
c               idata     integer(256,256)
c                         The original image data.
c
c               higram    integer(256)
c                         The histogram of the original
c                         image data.  The values for
c                         this array are passed back from
c                         the subroutine HIST.
c
c               hi        integer
c                         Highest gray level present in
c                         the original image.  Obtained
c                         from the HIST subroutine.
c
c               lo        integer
c                         Lowest gray level present in
c                         the original image.  Obtained
c                         from the HIST subroutine.
c
c               ty        integer
c                         Equalization option to execute.
c                         1 selects histogram stretching.
c                         2 selects histogram
c                         equalization.
c
c       Argument(s) supplied to the calling routine
c               edata     integer(256,256)
c                         The equalized image.
```

```
c
c       Routine(s) called by this routine
c               None.
c
c       Author
c               Keith W. Hefty
c
c       Date
c               22 June 1988
c
c       Revisions
c               20 July 1988  changed to 256 by 256 images
c
c**********************************************************
c
        subroutine equal (idata,edata,higram,hi,lo,ty)
c
c       Stores equalized image of idata into edata.
c       When ty equals 1 it stretches the histogram.
c       When ty equals 2 it equalizes the histogram.
c
        integer idata(256,256),edata(256,256),higram(256),
     &  sgram(256),row,col,ty,hi,lo,i
c
c       Option 1 - Gray levels are linearly mapped from
c       the minimum to maximum gray level range of the
c       orginal image to the gray level range (0 255).
c
        if(ty.eq.1)then
           print *,hi,lo
           do row=1,256
              do col=1,256
                 edata(row,col)=255*(idata(row,col)-lo)/
     &           (hi-lo)
              end do
           end do
        end if
c
c       Option 2 - Equalizes histogram.
c
c       Calculate mapping between orginal gray levels and
c       equalized gray levels.
c
        if(ty.eq.2)then
           sgram(0)=0
           do i=1,256
              sgram(i)=sgram(i-1)+higram(i)
           end do
           do i=1,256
```

```
            sgram(i)=255*sgram(i)/sgram(256)
         end do
c
c       Map orginal gray levels to equalized gray levels.
c
         do row=1,256
            do col=1,256
               edata(row,col)=sgram(idata(row,col))
            end do
         end do
      end if
c
      return
      end
```

```
C*****************************************************************
c
c       Subroutine
c               ENDO
c
c       Filename
c               ENDO.FOR
c
c       Calling sequence
c               call endo (array,tdata,uth,lth)
c
c       Description
c               Endo isolates areas of endosperm or bran
c               in a image based upon gray levels.  By
c               trial and error the user determines the
c               range of gray levels that represent either
c               bran or endosperm particles.  The user
c               enters this range at the prompt appearing
c               in the main program just before this
c               routine is called.
c
c       Argument(s) required from the calling routine
c               array    integer(256,256)
c                        An image in which the endosperm
c                        or bran particles are to be
c                        isolated.
c
c               uth      integer
c                        The upper bound on the range of
c                        gray levels that represent
c                        endosperm or bran particles.
c
c               lth      integer
c                        The lower bound on the range of
c                        gray levels that represent
c                        endosperm or bran particles.
c
c       Argument(s) supplied to the calling routine
c               tdata    integer(256,256)
c                        The result image with the
c                        endosperm or bran particles
c                        isolated.
c
c       Routine(s) called by this routine
c               None.
c
c       Author
c               Keith W. Hefty
c
```

```
c       Date
c               5 August 1988
c
c       Revisions
c               None.
c
c**********************************************************
c
        subroutine endo (array,tdata,uth,lth)
c
c       Double thresholds a 256 by 256 image.
c
        integer row,col,uth,lth,array(256,256),
     &  tdata(256,256),sum
c
c       Isolate areas of endosperm or bran, pixel with
c       grey levels outside of the upper and lower
c       threshold range are set to zero, while pixel with
c       grey levels with in the threshold range retain
c       their orginal grey level value.
c
        sum=0
        do row=1,256
           do col=1,256
              if(array(row,col).lt.lth.or.
     &            array(row,col).gt.uth)then
                 tdata(row,col)=0
              else
                 tdata(row,col)=array(row,col)
                 sum=sum+1
              end if
           end do
        end do
c
c       Print the number of total pixels found within the
c       upper and lower threshold range.
        print *,sum
c
        return
        end
```

```
C*****************************************************************
c
c       Subroutine
c               AREA
c
c       Filename
c               AREA.FOR
c
c       Calling sequence
c               area (idata,filename)
c
c       Description
c               Area counts the number of particles
c               isolated, calculates the size of each
c               particle, and computes the average gray
c               level of each particle.  All data
c               calculated is written to a file.  This
c               file serves as an input file to the
c               companion program TEST which summerizes
c               the findings in a brief format.
c
c       Argument(s) required from the calling routine
c               idata           integer(256,256)
c                               An image in which either the
c                               endosperm or bran particles
c                               have been isolated by the
c                               ENDO routine.
c
c               filename        character*20
c                               Filename to write area data.
c                               An extension of ".ad" is
c                               assumed.
c
c       Argument(s) supplied to the calling routine
c               None.
c
c       Routine(s) called by this routine
c               None.
c
c       Author
c               Keith W. Hefty
c
c       Date
c               15 September 1988
c
c       Revisions
c               None.
c
C*****************************************************************
```

```
c
      subroutine area (idata,fname)
c
      integer idata(256,256),pc(1000),ags(1000),pixcnt,
     &  gss,row,col,i,j,k,sp,stack(1000,2)
      character*20 fname
c
c     Initialize number of particles counter.
c
      k=1
c
c     Find a particle by scanning the image data array
c     until a nonzero gray level is found.  Test all
c     neighboring pixels for nonzero gray levels.  For
c     each nonzero neighbor increment the particle size
c     counter, add its gray level to the accumlated gray
c     level of the particle, and test all neighbors for
c     nonzero gray level.  After all pixels in a
c     particle have been found the averge gray level is
c     calculated and stored along with the size of the
c     particle in pixels into array.  The image array
c     is scanned for the next particle.
c
      do row=1,256
         do col=1,256
            if(idata(row,col).ne.0)then
               sp=0
               pixcnt=1
               gss=idata(row,col)
               idata(row,col)=0
               i=row
               j=col
30             if(idata(i,j+1).ne.0)then
                  pixcnt=pixcnt+1
                  gss=gss+idata(i,j+1)
                  idata(i,j+1)=0
                  if(j+1.lt.256)then
                     sp=sp+1
                     if(sp.gt.1000)then
                        print *,'ERROR'
                        print *,'subroutine  area'
                     end if
                     stack(sp,1)=i
                     stack(sp,2)=j+1
                  end if
               end if
               if(idata(i+1,j+1).ne.0)then
                  pixcnt=pixcnt+1
                  gss=gss+idata(i+1,j+1)
```

```
        idata(i+1,j+1)=0
        if(i+1.lt.256.and.j+1.lt.256)then
           sp=sp+1
           if(sp.gt.1000)then
              print *,'ERROR'
              print *,'subroutine area'
           end if
           stack(sp,1)=i+1
           stack(sp,2)=j+1
        endif
     end if
     if(idata(i+1,j).ne.0)then
        pixcnt=pixcnt+1
        gss=gss+idata(i+1,j)
        idata(i+1,j)=0
        if(i+1.lt.256)then
           sp=sp+1
           if(sp.gt.1000)then
              print *,'ERROR'
              print *,'subroutine area'
           end if
           stack(sp,1)=i+1
           stack(sp,2)=j
        end if
     end if
     if(idata(i+1,j-1).ne.0)then
        pixcnt=pixcnt+1
        gss=gss+idata(i+1,j-1)
        idata(i+1,j-1)=0
        if(i+1.lt.256.and.j-1.gt.1)then
           sp=sp+1
           if(sp.gt.1000)then
              print *,'ERROR'
              print *,'subroutine area'
           end if
           stack(sp,1)=i+1
           stack(sp,2)=j-1
        end if
     end if
     if(sp.ne.0)then
        i=stack(sp,1)
        j=stack(sp,2)
        sp=sp-1
        goto 30
     end if
     if(pixcnt.ne.0)then
        pc(k)=pixcnt
        ags(k)=gss/pixcnt
        k=k+1
```

```
               end if
            end if
         end do
      end do
      print *,k
c
c     Write area characteristics to a file.
c
      open (unit=5,file=fname,defaultfile='.ad',
     & status='NEW')
      write(5,99) k-1
      do i=1,k-1
         write(5,100) pc(i),ags(i)
      end do
      close(unit=5)
99    format (1x,i5)
100   format (1x,2i5)
      end
```

```
C*********************************************************
c
c       Main-routine
c               TEST
c
c       Filename
c               TEST.FOR
c
c       Execution sequence
c               run test
c
c       Description
c               Test is a companion program to the
c               ISOAREA program.  Whenever option 11, AREA
c               routine, of the ISOAREA program is
c               executed it writes a file containing the
c               data about the bran or endosperm particles
c               isolated using option 10, ENDO routine, of
c               the ISOAREA menu.  TEST takes this file,
c               which can contain up to 1000 lines and
c               condenses the data into four features.
c               These features are number of areas
c               isolated, average size, in pixels, of the
c               isolated areas, average gray level of the
c               isolated areas, and the ratio of the total
c               area covered by the isolated particles to
c               the total area in the image.
c
c       Argument(s) required from the calling routine
c               None.
c
c       Argument(s) supplied to the calling routine
c               None.
c
c       Routine(s) called by this routine
c               None.
c
c       Author
c               Keith W. Hefty
c
c       Date
c               4 October 1988
c
c       Revisions
c               None.
c
C*********************************************************
c
        integer i,k,c(1000),g(1000),sumsize,sumgl,avgsize,
```

```
     &  avggl,ratio
        character*20 name
c
c       Get file name from user.  The input file for this
c       program is written when option 11 of the ISOAREA
c       menu is executed.
c
        print *,'Enter filename > '
        accept 10,name
10      format(a)
c
c       Read in area data from file.
c
        open (unit=5,file=name,defaultfile='.ad',
     &  status='OLD')
        read (5,20) k
        do i=1,k
           read (5,25) c(i),g(i)
        end do
        close(unit=5)
20      format(1x,i5)
25      format(1x, 2i5)
c
c       Calculate average size and grey level of the
c       areas.
c
        sumsize=0
        sumgl=0
c
        do i=1,k
           sumsize=sumsize+c(i)
           sumgl=sumgl+g(i)
        end do
c
        avgsize=sumsize/k
        avggl=sumgl/k
c
c       Calculate ratio of areas to total frame area.
c
        ratio=100*sumsize/(256*256)
c
c       Print results of analysis to screen.
c
        print *,'Condensed data for file:', name
        print *,'  total number of areas:', k
        print *,'  average area size:', avgsize
        print *,'  average grey level:', avggl
        print *,'  percent of area to total frame:', ratio
c
```

```
end
```

Appendix B. Supporting Data

Feature	Roll	Endosperm open	Endosperm stan.	Endosperm close	Bran open	Bran stan.	Bran close
number of particles detected in image	1BK	-- 320	-- 320	-- 320	-- 501	-- 501	-- 501
	2BK	582 277	660 334	617 233	613 765	582 --	553 582
	3BK	665 373	825 338	-- 346	590 633	693 618	-- 857
average size of particles in pixels	1BK	-- 13	-- 13	-- 13	-- 75	-- 75	-- 75
	2BK	25 29	22 28	19 45	55 37	60 --	69 56
	3BK	15 18	12 18	-- 15	65 64	51 69	-- 35
average gray level of particles	1BK	-- 185	-- 185	-- 185	-- 147	-- 147	-- 147
	2BK	175 188	176 187	175 188	124 144	121 --	123 145
	3BK	174 187	171 188	-- 187	121 146	124 146	-- 150
ratio of particle area to image ar.	1BK	-- 6	-- 6	-- 6	-- 57	-- 57	-- 57
	2BK	22 12	22 14	18 16	51 43	53 --	58 50
	3BK	15 10	15 9	-- 7	58 62	54 65	-- 46

Top numbers are for Newton.
Bottom numbers are for Caldwell.
Stock sampled above rolls

Table B.1 HRW vs. SRW Test Results on Buhler Mill.

		Endosperm			Bran		
Feature	Roll	open	stan.	close	open	stan.	close
number of particles detected in image	1BK	221	206	198	620	472	524
	2BK	365	250	277	611	491	574
	3BK	304	192	237	639	548	527
average size of particles in pixels	1BK	21	19	15	58	87	18
	2BK	18	23	28	65	88	67
	3BK	17	15	20	63	81	81
average gray level of particles	1BK	193	192	192	144	142	139
	2BK	192	192	193	149	146	148
	3BK	197	191	192	150	141	144
ratio of particle area to image ar.	1BK	7	5	4	55	62	64
	2BK	10	8	12	61	66	59
	3BK	8	4	7	61	68	65

Stock sampled above rolls

Table B.2 Results from KSU Pilot Mill Test.

Feature	Roll	Endosperm open	Endosperm stan.	Endosperm close	Bran open	Bran stan.	Bran close
number of particles detected in image	1BK	286 352	206 352	229 352	567 645	472 645	509 645
	2BK	239 273	250 374	235 423	588 648	491 724	587 698
	3BK	282 479	192 479	348 344	621 725	548 552	632 588
average size of particles in pixels	1BK	14 16	19 16	22 16	66 52	87 52	71 52
	2BK	17 19	23 14	21 14	67 52	88 51	68 53
	3BK	12 11	15 14	13 11	66 56	81 80	66 73
average gray level of particles	1BK	196 185	192 185	197 185	149 147	142 147	151 147
	2BK	196 188	192 187	196 188	148 146	146 146	150 148
	3BK	196 187	191 186	191 186	148 148	141 153	151 146
ratio of particle area to image ar.	1BK	6 8	5 8	8 8	57 52	62 52	55 52
	2BK	6 7	8 8	7 9	60 51	66 56	61 56
	3BK	5 8	4 10	7 6	62 61	68 68	64 66

Top numbers are for Pilot Mill.
Bottom numbers are for Buhler Mill.
Stock sampled above rolls

Table B.3 Buhler/Pilot Mill Comparison Results.

Appendix C. Images of Mill Stock

Figure C.1 Image of 1BK Stock.

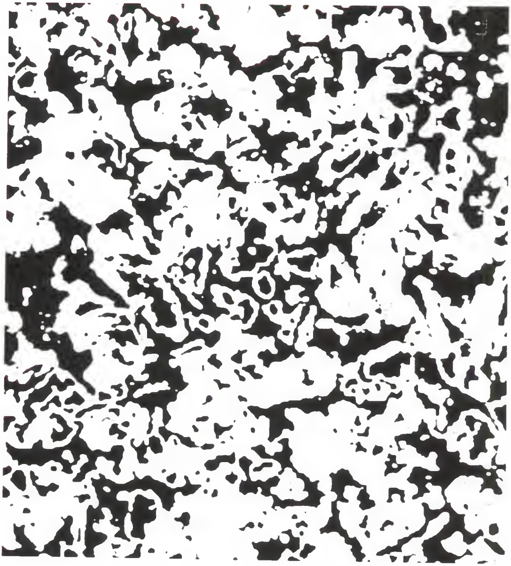


Figure C.2 Image With Endosperm Particles Isolated.

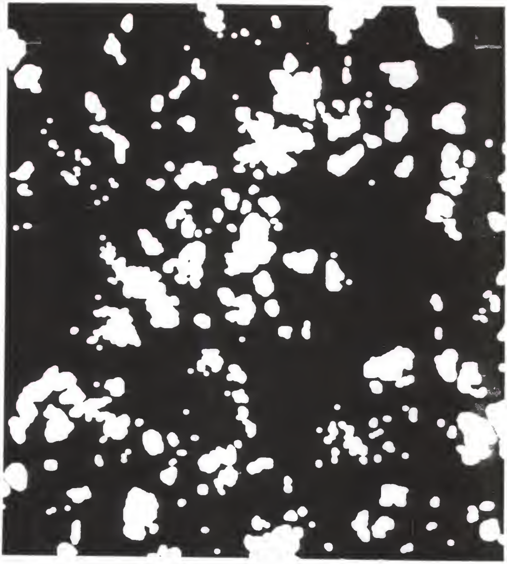


Figure C.3 Image With Bran Particles Isolated.

Optical Sensing as a Knowledge-Based System Input to Determine Flour Mill Break Stream Quality

by

Keith William Hefty

B.S., Kansas State University, 1988

AN ABSTRACT OF A THESIS

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Electrical Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1989

Abstract

The main goal of the work described is to determine the feasibility of using image processing technology to determine the quality of the break streams in a flour mill. The information obtained from the image analysis will be used as input information into a knowledge-based system that will control the break roll gaps. A presentation of the knowledge representation for a prototype knowledge-based system is given.

The results suggest that break stream quality can be detected by optical analysis, but the use of suggested improvements in sampling handling procedures and image processing techniques should be incorporated to increase the reliability of results obtained.