Research Technical Completion Report
A-081-IDA

COMPUTER-LINKED CCD CAMERA FOR SEDIMENT SHAPE ANALYSIS

by

K.A. Prisbrey
Associate Professor of Metallurgy
College of Mines and Earth Sciences

R.E. Rinker
Associate Professor of Computer Science
College of Engineering

M.B. Aboukheshem
Graduate Student
College of Mines and Earth Sciences

Idaho Water and Energy Resources Research Institute
University of Idaho
Moscow, ID 83843

March 1983

The research on which this report is based was financed in part by the U.S. Department of the Interior, as authorized by the Water Research and Development Act of 1978 (P.L. 95-467).

Research Technical Completion Report
A-081-IDA

COMPUTER-LINKED CCD CAMERA FOR SEDIMENT SHAPE ANALYSIS

by

K.A. Prisbrey
Associate Professor of Metallurgy
College of Mines and Earth Sciences

R.E. Rinker
Associate Professor of Computer Science
College of Engineering

M.B. Aboukheshem
Graduate Student
College of Mines and Earth Sciences

Submitted to
United States Department of the Interior
Washington, D.C. 20242

The work on which this report is based was supported in part by funds provided by the United States Department of the Interior as authorized under the Water Research and Development Act of 1978.

Idaho Water and Energy Resources Research Institute
University of Idaho
Moscow, ID 83843

March 1983

ACKNOWLEDGEMENTS

We gratefully acknowledge OMARK Industries, Lewiston, Idaho for the use of their CCD camera and electronics. Also for funding provided by the Department of the Interior through the Idaho Water and Energy Resources Research Institute, and through the Mining and Mineral Research Institute.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

A major problem with using chemical and mineralogical composition analysis to trace stream sediment origins is the cost. A solution to this problem is to use Fourier shape descriptors of individual sediment particles, which gives an inexpensive means of distinguishing sediments even with similar chemical and mineralogical characteristics. A CCD camera was linked to a PDP 11 computer to provide a rapid method of data gathering for such analysis.

INTRODUCTION

The value of establishing upstream sources of stream sediments has been realized for many years. But in the past this required expensive chemical or mineralogical analysis, and expertise often not available. Furthermore, chemical and mineralogical data are often not sufficient.

Particle shape can, when measured very accurately, distinguish between particle groups that are chemically or mineralogically the same. Shape contains an implicit formational and post-formational history, with unique parameters for any given particle set. For this reason, particle shape analysis is a valuable analytical tool for rapid and inexpensive tracing of stream sediments.

Once sediments have been uniquely identified, their source can be established. The value of these techniques is for general water resource management applications such as dam location, dam siltation, fish habitat studies, and sediment transport modeling.

Our objective of using particle shape as a tracer has already proven to be valuable in closely related disciplines. Ehrlich et al., (6), using quartz as a tracer, were able to determine the relative accumulation of river, beach and cliff sands at Oceanside, California. The three types of sand were clearly defined based on fifty-seven samples of about one hundred grains per sample. Results indicated that tidal currents regularly removed river sand from the beach during dry periods of low river flow. At this time the dominant source of sand was from the cliffs located behind the beach.

Similar shape analysis confirmed the existence of an "energy fence" in ocean transport of beach sands. Two studies by Yarus (12) on the eastern coast of the Gulf of Alaska, and Brown et al. (3) on the southeastern United States continental shelf, found highly angular sands, attributed to fluvial sources, in the near shore zone, and higher percentages of smoother abraded sands further out on the shelf. Additional detailed flow patterns were discerned beneath the sea to extend the information about land formations. Each study used more than eighty samples with two hundred grains of fine sand per sample.

Mazzullo and Ehrlich (8) extended the application to explain the formation of the St. Peter sandstone. They located two transgressive phases by identifying a vertical pattern of two shape families. Before this the St. Peter sandstone was considered a single thick transgressive sheet of relatively homogeneous, featureless orthoquartizite. Using fifty eight samples of over two hundred grains per sample, they found a vertical pattern of two shape families (abraded vs angular), and attributed this finding to the movement of the littoral energy fence and its effect on sedimentation.

The shape technique has been recently applied to the tracking of abyssal silts in the insular rise of the south Iceland (7). Eighty samples containing two hundred grains per sample resulted in the definition of three shape families which are representative of three transport processes in the region: ice-rafting, thermohaline flow from the Norwegian Sea, and episodic turbidity currents.

Development of shape analysis techniques has been valuable to many related disciplines (7-11). However, the measurement of particle shapes has proven to be a time consuming and expensive process. Ehrlich (5) reported that a single sample of two hundred grains required four to six hours to hand draw and hand digitize, before he developed more rapid automated techniques. Our experience has shown considerably longer times required. Therefore we successfully interfaced a CCD (charge coupled device) camera to a PDP 11/23 computer, and mounted the camera on a microscope for rapid digitization of sediment particles. This technique allowed the easy transfer of the data to the campus main-frame computer for analysis using SAS computer language. Over 120 representative particle samples were digitized and the power of the SAS language to discriminate between shape classes was demonstrated. This report focuses on the development of the digitizing apparatus.

METHODS AND MATERIALS

One of the unique features of this system is the apparatus used to collect the data. In the past, a particle shape was entered into the computer by projecting the particle shape on a large piece of graph paper, then manually digitizing the points along the edge of the particle, and finally entering the x and y coordinates of each point into the computer. Since several hundred points were often involved in each particle this was a very laborious and time-consuming task. It usually took several hours to enter one single particle.

The next method used was to project the enlarged image onto an electronic digitizing tablet, and then, using a push-button cursor, points along the particle edge were entered. While the electronic tablet performed the conversion from raw positional information to x and y coordinate data, still each of the several hundred points needed to be manually entered. At best, this process would take an hour or more to perform.

The method used here involves the use of a digital television camera directly connected to the computer. The camera converts a picture into a 128 by 128 picture element matrix, with each picture element (pixel) having a value from 1 to 256, depending upon the brightness of the particle pixel. When a picture of a particle is taken the background shows up very bright, due to backlighting of the particle, while the particle itself appears as a dark spot in the picture. The structure of the camera itself produces the digitization of the particle shape.

The output for the camera is connected to the computer via a direct memory access (DMA) inteface. This arrangement effectively couples the digitized pixel values coming from the camera directly to the computer's memory, allowing the picture to be stored directly into the computer for processing. An entire picture can be taken and sent to the computer in less than a tenth of a second.

Once inside the computer, a computer program must analyze the picture and determine where the edge of the particle is. This is done by locating the boundary between the bright background and the dark particle shape. This rather complicated procedure can be performed in a few seconds. Thus, the data acquisition time drops from several hours, using a totally manual method, to a few seconds, using the computer-camera configuration.

The exact hardware used in this project is listed below:

PDP11/23 Computer with 256 kilobytes of memory and an RL01 disk drive and terminal.

General Electric PN2200 digital television camera and controller.

MDB Systems DRV11-B DMA inteface.

Peritck VCG-Q video display inteface and a Mitsubishi 19" color monitor.

The latter piece of equipment (the video interface and monitor) was used to display the particle pictures as it was sent to the computer. Figure 1 shows the interconnection between the equipment used.

Figure 2 shows an actual photograph of the entire system. The computer is on the right hand side, with the camera controller and its associated circuitry on top of the computer cabinet. The computer terminal is just to the left of the computer. The camera and stand are to the left of the terminal, and finally the color display monitor is on the extreme left.

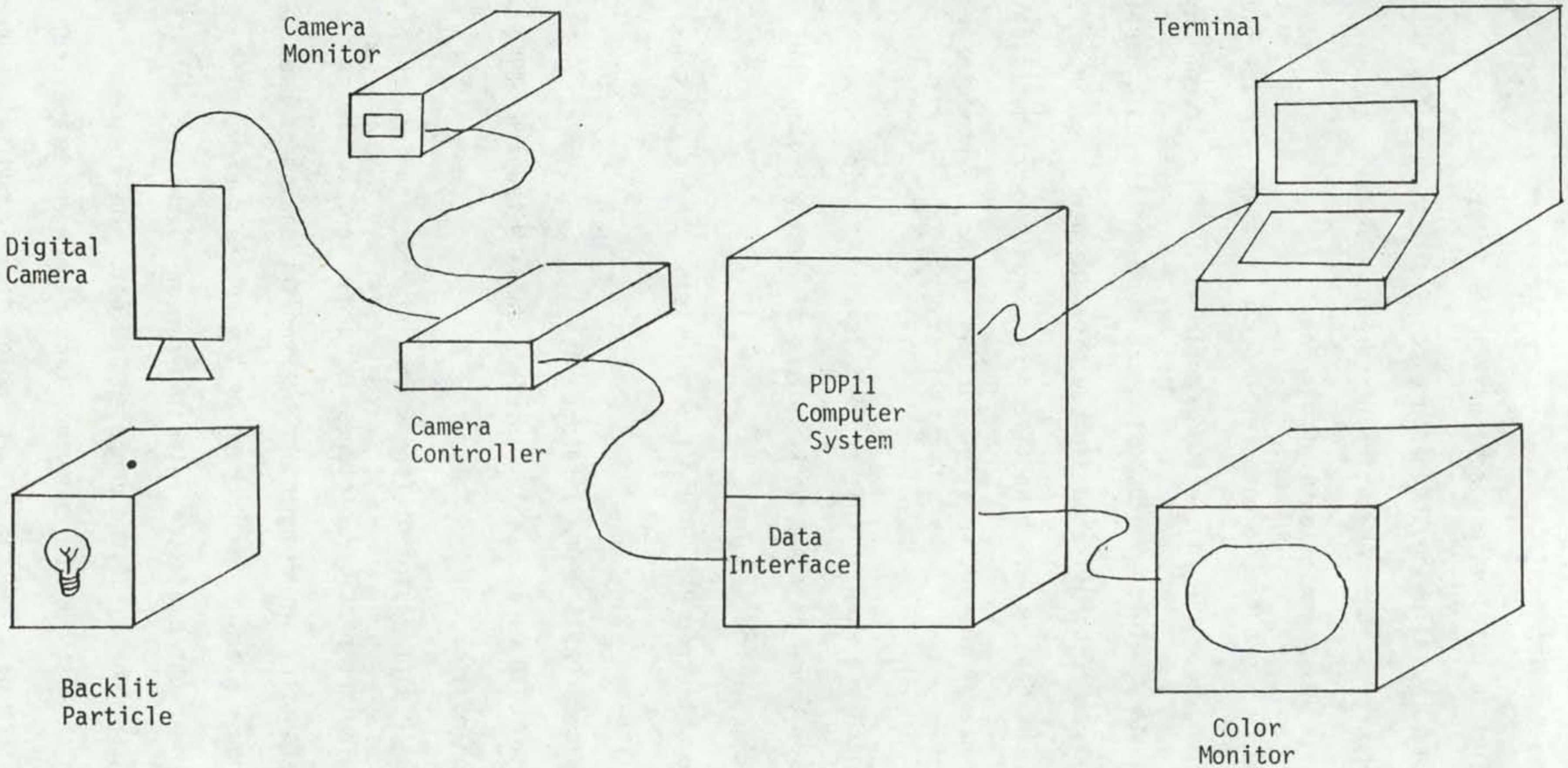

Figure 1. Apparatus setup for the computer driven particle shape detection system.

Figure 2. Actual System With Camera Controller And Associated Circuitry On Computer Cabinet To The Right.

Figure 3 is a close-up of the camera and camera stand. Figure 4 shows the camera controller and camera monitor. The latter piece of equipment is used to focus the camera before a picture is sent to the computer. The computer code, which takes a picture, determines the edge coordinates of the particle, and places these coordinates into a disk file, as listed in the Apendix.

A shakedown experiment was conducted using glass spheres. A large steel ball was dropped on 5 mm glass spheres producing fragments of different shapes from each glass sphere. The steel ball was dropped from three heights producing fragments associated with three energy levels of fracture.

These fragments were carefully collected and digitized under the camera apparatus. A centroid of the X, Y coordinates was found for each particle to allow conversion to polar coordinates. Fourier transforms of these polar coordinates were taken and used as the primary shape descriptors of each glass fragment. Fourier shapes descriptors were collected for over 120 glass fragments.

We tested the ability of the shape descriptors to uniquely classify each glass fragment to its associated fracture energy level by using a linear discriminant function. Possibly, there is something similar despite the variety of irregular shapes produced from the fracture of glass spheres at a single energy level. This is a severe test of the potential use of the shape as a tracer in future siltation studies. The test is also of theoretical significance to fracture mechanics and comminution theory.

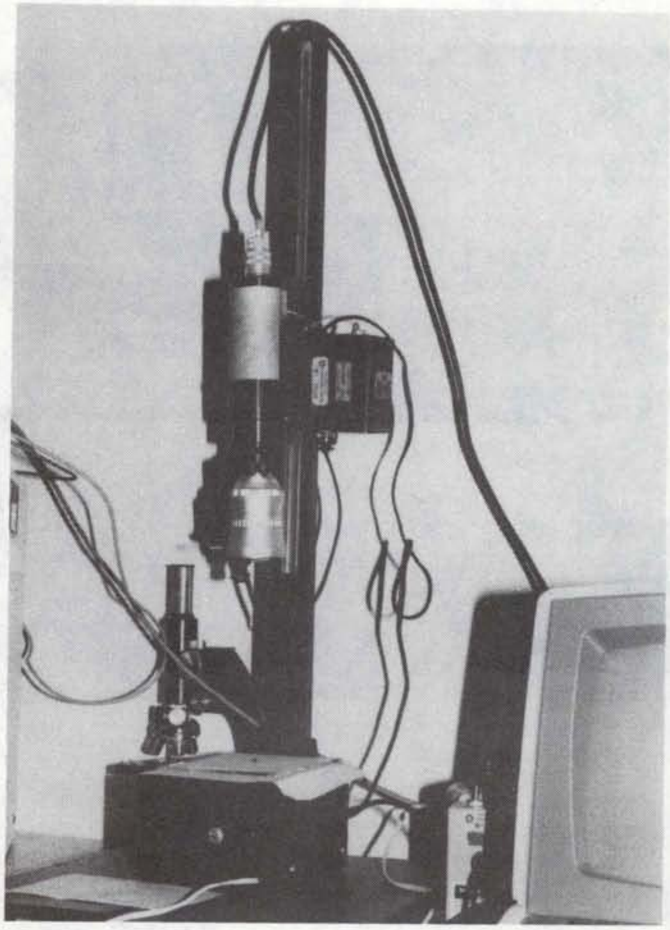

Figure 3. Close-up Of Camera And Camera Stand

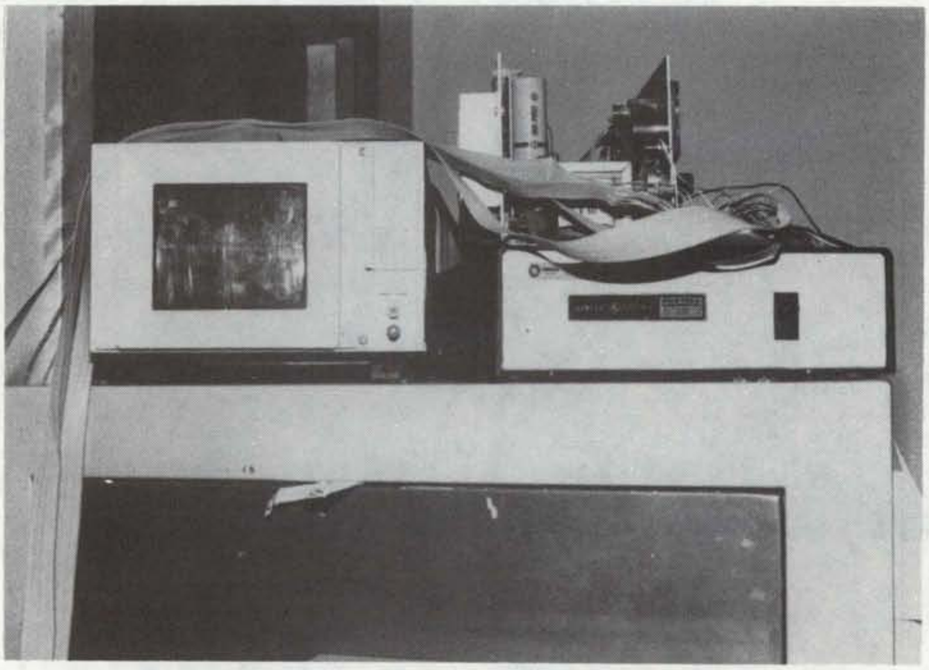

Figure 4. Close-up Of Camera Controller And Camera Monitor

RESULTS

Glass spheres were shattered by dropping a steel ball on them from three different heights. The fragments from each sphere were collected and subjected to shape analysis as a shakedown test for the shape measuring apparatus. Figure 5 shows a log-log plot for the Fourier coefficients vs. harmonic number for a single glass fragment. This "Meloy plot" is unique for each fragment. Table 1 shows the results of using only 19 Fourier coefficeints to correctly classify over 120 glass fragments. Table 2 shows even better results by increasing to 40 Fourier coefficients. Particles can be correctly associated with their energy group with over 80% accuracy. These results show the ability of shape measurements to be an effective tracer in siltation studies, and to be of use in other particulate systems.

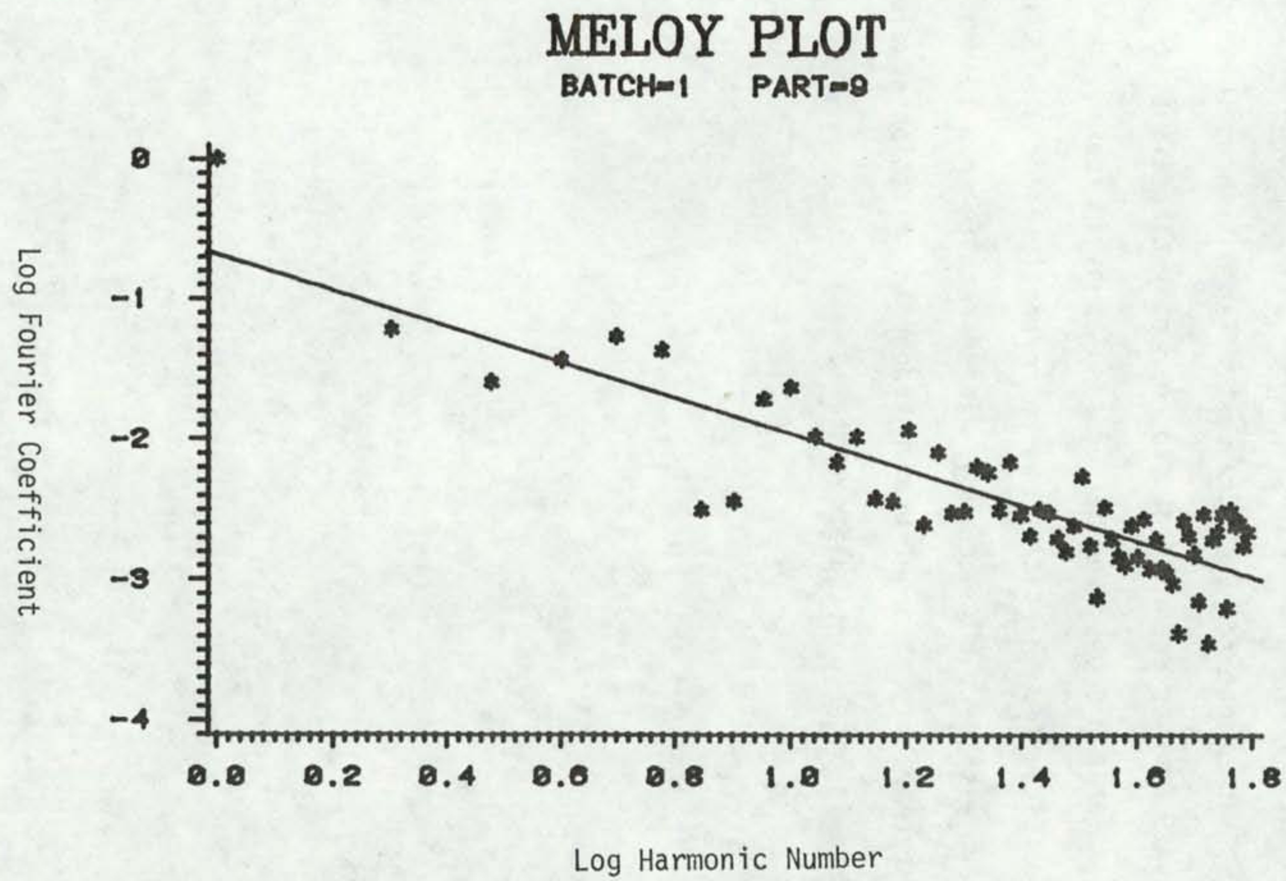

Figure 5. Lead Particles

Table 1. Percent of 120 Glass Fragments
Classified to Correct Groups
(Energy Level of Production)
Using 19 Fourier Coefficients
In a Linear Discriminant Function.

Energy Level	High	Medium	Low
High	70	10	20
Medium	18	65	17
Low	20	13	67

Table 2. Percentage of 120 Glass Fragments
Classified to Correct Groups
(Energy Level of Production)
Using 40 Fourier Coefficients
In a Linear Discriminant Function.

Energy Level	High	Medium	Low
High	90	5	5
Medium	10	80	10
Low	5	8	87

DISCUSSION AND CONCLUSIONS

Digitizing a group of particles and extracting their Fourier shape descriptors is now a much faster process. Larger groups of particles can be conveniently digitized and analyzed according to Fourier shape descriptors. We conclude that:

1) Particle groups can be distinguished by Fourier shape descriptors,
2) the technique has a wide variety of research applications,
3) a useful, low-cost tracer technique has been developed for stream sediments, and
4) a CCD camera coupled to a computer provides an efficient digitizing method.

REFERENCES

1. Allen, J.R.L. 1970. Physical Processes of Sedimentation. Elsevier, New York, 248 p.

2. Beddow, J.K. 1980. "Particulate Morphology Data", Advanced Particulate Morphology. ed. J.K. Beddow, T.P. Meloy, Boca Raton, Florida. CRC Press, pp. 1-84.

3. Brown, P.J., R. Erhlich, and D.J. Colquhoun. 1980. "The Origin of Patterns of Quartz Sand Types on the Southeastern U.S. Continental Shelf." Jour. Sed. Petrology, in Press.

4. Ehrlich, R. "Analysis of Particle Morphology Data", Advanced Particulate Morphology, ed. J.K. Beddow, T.P. Meloy, Boca Raton, Florida. CRC Press pp. 101-119.

5. Ehrlich, R. Automated Particle Shape Analysis-Development of a microprocessor controlled image analysis system. Computer and Geosciences, to be published.

6. Ehrlich, R., J.J. Orzeck, and B. Weinberg. 1974. Detritial quartz as a natural tracer-Fourier Grain Shape Analysis. Jour. Sed. Petrology, vol. 44, no. 1, p. 145-150.

7. Kellog, T.B. 1976. Late Quaternary climatic changes; evidence from deep-sea cores of Norwegian and Greenland seas. Geol. Soc., America Mem., 145, p. 77-110.

8. Mazzullo, J. and R. Ehrlich. 1980. A vertical pattern of variation in the St. Peter Sandstone - Fourier Grain Shape Analysis. Jour. Sed. Petrology, vol. 50, no. 1, p 63-70.

9. Powers, M.C. 1953. "A New Roundness Scale for Sedimentary Particles". J. of Sedimentary Petrology, vol. 23, pp. 117-119. 1969.

10. Scharz, H.P. and K.C. Shane. "Measurement Particle Shape by Fourier Analysis". Sedimentology, vol. 13, pp. 213-231, 1969.

11. Wadell, Hakon. 1932. "Volume Shape of Roundness of Rock Particles". J. of Geology, vol. 40, pp. 443-451.

12. Yarus, J.M. 1978. Shape variation of sand sized quartz in the eastern gulf of Alaska and its hydrodynamic interpretration - Fourier Grain Shape Analysis. (Unpub. Ph.D. thesis) Univ. of So. Carolina, Columbia, S.C.

APPENDIX

```
      PROGRAM CAMERA
C ***********************************************************************
C *                                                                     *
C *                    CAMERA READING PROGRAM                           *
C *                                                                     *
C *   This is the main program for the camera reading and edge          *
C *   finding system. First it initializes the entire system,           *
C *   reads the camera (takes a picture), and then calculates           *
C *   some statistics about the picture. It then allows the user        *
C *   to either take another picture or analyze the picture that        *
C *   has been taken. To analyze the current picture, an edge-          *
C *   finding algorithm is employed, after which the edge points        *
C *   are placed into a file for further analysis.                      *
C *                                                                     *
C *   Routines Called by this program:                                  *
C *                                                                     *
C *      INITCOL - Does the initialization of the color display         *
C *      GETBLK  - Actually gets a block of data from the camera        *
C *                (takes the picture). This routine is written         *
C *                in PDP11 assembly language.                          *
C *      CONVRT  - Reformats the raw picture data into a usable,        *
C *                2D array form.                                       *
C *      STATS   - Finds the high, low, and average value of the        *
C *                pixel data.                                          *
C *      FNDEDG  - Routine which actually finds the edge of the         *
C *                particle.                                            *
C *                                                                     *
C ***********************************************************************

      INTEGER SUM,AVE,HIGH,LOW
      BYTE RAWPXL(17100),ARRAY(127,128)
C
      CALL INITCOL                       !sets up necessary colors
5     CALL GETBLK(17018,RAWPXL)          !takes a picture
      CALL CONVRT(RAWPXL,ARRAY)          !puts raw pixel data into 2d array
C
C     This next part prints out a crude representation of the entire
C     pixel array. It was first put in for diagnostic purposes, but seems
C     useful for getting a ballpark idea of what's going on
C
      DO 1 I=1,127,2
      WRITE(7,2)(ARRAY(I,J)/4,J=1,128,2)
2     FORMAT(1X,64I1)
1     CONTINUE
C
      CALL STATS(ARRAY,SUM,AVE,HIGH,LOW) !finds statistics of pixel data
C
C     Ths threshhold level calculated below was to be used to help
C     automate the edge-finding process. In fact, the value it calculates
C     is usually not the best threshhold level for the edge. Still, it
C     does provide useful information and was left in.
C
      ITHRESH=(AVE+LOW)/2
      WRITE(7,3)AVE,HIGH,LOW,ITHRESH
3     FORMAT(1X,'AVE =',I3,10X,'HIGH =',I3,10X,'LOW =',I3,10X,
     *'THRESHHOLD =',I3)
C
      WRITE(7,*)'WANT ANOTHER PICTURE?'
      READ(5,4)ANS
4     FORMAT(A1)
      IF(ANS.EQ.'Y')GO TO 5
C
      WRITE(7,*)'ENTER THRESHHOLD VALUE'
      READ(5,10)ITHRESH
10    FORMAT(I)
      CALL FNDEDG(ARRAY,ITHRESH)         !edge finding routine
      STOP
      END
```

```
      SUBROUTINE INITCOL
C ***********************************************************************
C *                                                                     *
C *    This routine initializes the colors to be displayed on the       *
C *    color raster display. Actually what we want is to be able to     *
C *    display gray levels, so the various "colors" are set to          *
C *    various gray scales. Color number zero is set to RED, and this   *
C *    color is used to outline the edge on the screen when it is       *
C *    found by the edge-finding routine.                               *
C *                                                                     *
C ***********************************************************************

C     PLOTS and SETCOL are routines in our plotting library. PLOTS is
C     the plot initialization routine, while SETCOL sets the color number
C     specified in the first argument to the RED, GREEN, and BLUE values
C     specified in the next arguments (0 is off for that color, 15 is full
C     ON for that color)
      CALL PLOTS(3,0,10)
      DO 100 I=1,14
100      CALL SETCOL(I,I+1,I+1,I+1)
C
C     Set color number zero to full red:
C
      CALL SETCOL(0,15,0,0)
      RETURN
      END
```

```
      SUBROUTINE CONVRT(RAWPXL,ARRAY)
C *********************************************************************
C *                                                                   *
C *     This routine takes the raw pixel data as received from        *
C *     GETBLK and reformats it into a 2d array of the picture        *
C *     The raw data comes from the camera as just a large string     *
C *     (1D array) of pixel values, with several dummy pixel values   *
C *     at the end of each scan line. These are used to determine if  *
C *     the correct number of pixels have been received for each line *
C *     and are then stripped off.                                    *
C *                                                                   *
C *********************************************************************

      BYTE RAWPXL(17100),ARRAY(127,128)
      IPXL=3
C
      DO 1 IROW=1,127
         DO 2 I=1,10
            IF(RAWPXL(IPXL).GE.0)GO TO 100
              IPXL=IPXL+1
2        CONTINUE
         STOP 'TOO MANY EOL PIXELS'
100      DO 101 ICOL=1,128
            IF(RAWPXL(IPXL).LT.0)STOP'FOUND AN EOL PIXEL IN WRONG PLACE'
              ARRAY(IROW,ICOL)=RAWPXL(IPXL)
              CALL WRPIXL(IROW,ICOL,(ARRAY(IROW,ICOL)-1)/4)
              IPXL=IPXL+1
101      CONTINUE
1     CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE STATS(ARRAY,SUM,AVE,HI,LOW)
C ***********************************************************************
C *                                                                     *
C *    This routine calculates the high, low and average pixel values  *
C *    within the pixel array. These statistics are used to determine  *
C *    The threshhold level used in the edge-finding routine.          *
C *                                                                     *
C ***********************************************************************
      BYTE ARRAY(127,128)
      INTEGER SUM,AVE,HI,LOW,ROWAVE,ROWSUM
      HI=-1
      LOW=127
      SUM=0
C
      DO 1 IROW=1,127
         ROWSUM=0
         DO 2 ICOL=1,128
            ROWSUM=ROWSUM+ARRAY(IROW,ICOL)
            IF(ARRAY(IROW,ICOL).GT.HI)HI=ARRAY(IROW,ICOL)
            IF(ARRAY(IROW,ICOL).LT.LOW)LOW=ARRAY(IROW,ICOL)
2        CONTINUE
         ROWAVE=ROWSUM/128.+.5
         SUM=SUM+ROWAVE
1     CONTINUE
C
      AVE=SUM/127.+.5
C
      RETURN
      END
```

```
      SUBROUTINE FNDEDG(ARRAY,ITHRESH)
C ************************************************************************
C *                                                                      *
C *     This routine finds the edge of the particle within the picture  *
C *     array. It first goes half way up the picture, then scans across *
C *     the picture until it finds the first pixel in the edge of the   *
C *     particle (actually, it scans across until it finds a pixel with *
C *     a value less than the threshhold level). It then proceeds in    *
C *     clockwise direction, looking for the next edge pixel.           *
C *                                                                      *
C *     This routine works well if the edge of the particle is very     *
C *     well-defined, with good contrast between the particle boundary  *
C *     and the background color. This is not always the case, and so   *
C *     it is possible for the routine to fail. In this case, the       *
C *     simplest thing is done - the routine stops. This is where some  *
C *     real improvement could be made - make this routine handle the   *
C *     situation better. This would be essential if this were to be    *
C *     used in an industrial situation.                                 *
C *                                                                      *
C ************************************************************************

      BYTE ARRAY(127,128),MOVEX(8),MOVEY(8)
      INTEGER PRSNTX,PRSNTY,NXTRYX,NXTRYY,FIRSTX,FIRSTY
C
C     Each of the eight possible directions we could move from a given
C     point to another adjacent point is assigned a number  from 1 to 8
C     This array contains the relative X and Y movements needed to move
C     To the particular point.
C
      DATA MOVEX(1),MOVEY(1)/-1,1/
     *     MOVEX(2),MOVEY(2)/0,1/
     *     MOVEX(3),MOVEY(3)/1,1/
     *     MOVEX(4),MOVEY(4)/1,0/
     *     MOVEX(5),MOVEY(5)/1,-1/
     *     MOVEX(6),MOVEY(6)/0,-1/
     *     MOVEX(7),MOVEY(7)/-1,-1/
     *     MOVEX(8),MOVEY(8)/-1,0/
C
C   Open file where coordinates are to go
C
      WRITE(7,1)
1     FORMAT(' ENTER FILE NAME'/)
      CALL ASSIGN(1,,-1)
C
C    First, find starting point. Go to middle row, and start scanning
C    across until we find the object edge.
C
      DO 10 ICOL=3,125  !These limits will miss 1st & last column garbage
         IF(ARRAY(64,ICOL).LE.ITHRESH) GO TO 20
10    CONTINUE
      STOP'NO OBJECT FOUND'
C
C   We get here if the edge is found. the next statements will cause the
C   edge search to proceed "up" (clockwise) from the present position
C
20    LASTMV=2
      NUMPTS=0
      FIRSTX=64
      FIRSTY=ICOL
      PRSNTX=64
      PRSNTY=ICOL
```

```
C
C   Top of the main loop
C
30       WRITE(1,31)PRSNTX,PRSNTY
31       FORMAT(2I4)
         CALL WRPIXL(PRSNTX,PRSNTY,0)
         NUMPTS=NUMPTS+1
         NXTRYX=PRSNTX+MOVEX(LASTMV)
         NXTRYY=PRSNTY+MOVEY(LASTMV)
         IF(ARRAY(NXTRYX,NXTRYY).LE.ITHRESH)GO TO 40
C
C   We come here if our next try at moving didn't find a point on the
C   object itself. We then start looking, in a clockwise direction
C   (that is, increment the "last-move" index), for the edge. When we
C   find it, it becomes the next point considered to be on the edge.
C
         IMOVE=LASTMV+1
         IF(IMOVE.GT.8)IMOVE=1
35       NXTRYX=PRSNTX+MOVEX(IMOVE)
         NXTRYY=PRSNTY+MOVEY(IMOVE)
         IF(ARRAY(NXTRYX,NXTRYY).LE.ITHRESH) GO TO 50 !We found it!
         IMOVE=IMOVE+1
         IF (IMOVE.GT.8)IMOVE=1
         IF(IMOVE.EQ.LASTMV)STOP 'Went full circle looking for edge 30'
         GO TO 35
C
C   We get here if our next try at moving found a point on the object.
C   We now need to see if the point is actually an edge point. We do
C   this by searching counterclockwise until we find a point NOT on
C   the edge.
C
40       IMOVE=LASTMV-1
         IF(IMOVE.LE.0)IMOVE=8
45       NXTRYX=PRSNTX+MOVEX(IMOVE)
         NXTRYY=PRSNTY+MOVEY(IMOVE)
         IF(ARRAY(NXTRYX,NXTRYY).GT.ITHRESH)GO TO 46 !Found it!
         IMOVE=IMOVE-1
         IF(IMOVE.LE.0)IMOVE=8
         IF(IMOVE.EQ.LASTMV)STOP 'Went full circle 40!'
         GO TO 45
C
46       IMOVE=IMOVE+1
         IF(IMOVE.GT.8)IMOVE=8
C
C   We have now found the next edge point. We make it the "present"
C   point (for the next "go-round.")
C
50       LASTMV=IMOVE
         PRSNTX=NXTRYX
         PRSNTY=NXTRYY
C
C   Check to see if we have gone completely around the object. We're
C   done if we have.
C
         IF(NUMPTS.LT.50)GO TO 30
         IF(IABS(PRSNTX-FIRSTX).GT.1.OR.IABS(PRSNTY-FIRSTY).GT.1)
     *                                                   GO TO 30
         CALL CLOSE(1)
         WRITE(7,7)NUMPTS
 7       FORMAT(1X,I5,' points written to the file')
         RETURN
         END
```

```
;
;       ROUTINE TO TRANSFER INFORMATION FROM A DMA INTERFACE TO A
;       FORTRAN ARRAY.
;
;       CALLING SEQUENCE:
;
;               CALL GETBLK(N,ARRAY)
;
;       WHERE:
;
;
;               N - NUMBER OF ELEMENTS TO TRANSFER
;               ARRAY - ARRAY WHERE BLOCK OF VALUES IS TO BE
;                       TRANSFERRED
;
;
;***********************************************************************
;
        WCR     = 172410
        BAR     = WCR+2
        CSR     = WCR+4
        IDBR    = WCR+6
        ODBR    = WCR+6
;
;
        .GLOBL  GETBLK

GETBLK: MOV     @2(R5),R0
        COM     R0
        MOV     R0,@#WCR        ; MOVE NUMBER OF ELEMENTS TO WORD COUNT REG
        MOV     4(R5),@#BAR     ;PUT STARTING ARRAY ADDR INTO BUS ADDR REG
        MOV     #6,@#CSR        ;INITIALIZE A00 F/F
        MOV     #2,@#CSR        ;THIS PRESETS A00 TO 1
        MOV     #6,@#CSR
WAIT:   MOV     @#CSR,R0        ;GET CSR VALUE
        BIT     #4000,R0        ;LOOK FOR EOF
        BEQ     WAIT            ;IF NOT THERE, JUST CONTINUE TO WAIT
        MOV     #7,@#CSR        ;IF EOF FOUND, SET GO BIT TO START DMA
WAIT1:  BIT     #200,@#CSR      ;IS TRANSFER DONE?
        BEQ     WAIT1           ;NO - LOOP BACK AND WAIT
        RTS     PC              ;YES - RETURN
        .END
```

Selected Water Resources Abstracts

Input Transaction Form

1. Report No.	2.	3. Accession No.
		W

4. Title

COMPUTER-LINKED CCD CAMERA FOR SEDIMENT SHAPE ANALYSIS

5. Report Date

6. March 1983

8. Performing Organization Report No.

7. Author(s)

Prisbrey, K.A., Rinker, R.E., Aboukheshem, M.B.

10. Project No.

A-081-IDA

9. Organization

Idaho University, Moscow, College of Mines and Earth Resources

11. Contract/Grant No.

14-34-0001-0216

13. Type of Report and Period Covered

12. Sponsoring Organization

15. Supplementary Notes

Idaho Water and Energy Resources Research Institute Completion Report, Moscow, March 1983, 28 p, 5 fig, 2 tab, 12 ref.

16. Abstract

A major problem with using chemical and mineralogical composition analysis to trace stream sediment origins is the cost. A solution to this problem is to use Fourier shape descriptors of individual sediment particles, which gives an inexpensive means of distinguishing sediments even with similar chemical and mineralogical characteristics. A CCD camera was linked to a PDP 11 computer to provide a rapid method of data gathering for such analysis.

17a. Descriptors

Image processing, Data collection, Computer Fourier analysis, Data acquisition, Shape analysis, Camera interfacing, Data interpretation

17c. COWRR Field & Group

08E
09C

18. Availability	19. Security Class. (Report) / 20. Security Class. (Page)	21. No. of Pages 28 / 22. Price $2.50	Send to: Water Resources Scientific Information Center OFFICE OF WATER RESEARCH AND TECHNOLOGY U.S. DEPARTMENT OF THE INTERIOR Washington, D.C. 20240

Abstractor IWERRI Institution IWERRI