**Abstract**

In this preliminary report, we present a computer program called MAT for implementing the PDQ factorization for digitized images of large pictorial data base. Within a mini-computer, particular emphasis has been placed on the study of the efficiency of the algorithm in the storage and process of the images.
## CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>I. INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>II. ANALYSIS OF PDQ FACTORIZATION</td>
<td>1</td>
</tr>
<tr>
<td>III. DESCRIPTION OF THE MATRIX FACTORIZATION SOFTWARE</td>
<td>4</td>
</tr>
<tr>
<td>IV. TEST RESULTS</td>
<td>4</td>
</tr>
<tr>
<td>V. PDQ FACTORIZATION FOR BLURRED IMAGES</td>
<td>4</td>
</tr>
<tr>
<td>VI. CONCLUDING REMARKS</td>
<td>6</td>
</tr>
<tr>
<td>VII. LIST OF PROGRAMS</td>
<td>6</td>
</tr>
<tr>
<td>ACKNOWLEDGEMENT</td>
<td>11</td>
</tr>
<tr>
<td>REFERENCES</td>
<td>1</td>
</tr>
<tr>
<td>ACCESSION FOR</td>
<td></td>
</tr>
<tr>
<td>NTIS GRA&amp;I/DVIC TAB Unannounced Justification</td>
<td></td>
</tr>
<tr>
<td>By Distribution/Availability Codes/Avail and/or Special</td>
<td></td>
</tr>
<tr>
<td>Dist A</td>
<td></td>
</tr>
</tbody>
</table>
ON DIGITAL PROCESSING VIA PDQ FACTORIZATION

I. Introduction

This report discusses a method of compressing data bases that describe large digital images. The ultimate objective of such a procedure is to effect savings in computer memory required to process images and to decrease the amount of data necessary to transmit images over a communication channel.

A digital picture (of photograph or video image) can be represented as a matrix consisting of rectangular blocks of size $m \times n$ where $m$ and $n$ are the number of pels (picture elements) in the horizontal and vertical directions, respectively. To process such a picture most effectively (in terms of processing time and quantities of the pictures) by a computer, it is desirable that the main memory of the computer, during the computation, has enough capacity for storage of the entries of the whole image matrix. Unfortunately, this is not the case for most modern minicomputers. Therefore, different adaptive techniques have been devised to cope with this situation. For instance, C. B. Moler and G. W. Stewart [2] described a matrix factorization, the so called PDQ factorization technique and applied it to image representation. The factorization is closely related to SVD (singular value decomposition technique developed by H. C. Andrews and C. L. Patterson [1]). The merit of PDQ factorization over SVD is that it requires, during computation, much less arithmetic and the entire matrix need not be stored in the main memory.

We have applied the PDQ factorization technique to various digitized pictures, and studied their approximations by different ranks with fixed initial column vector and observed the relationships between initial column vectors $P_1$ and resolution of the approximated image with fixed rank.

In the implementation of PDQ factorization, we have developed a computer program called MATPRXVAX written in Fortran. This program is described in some detail.

II. Analysis of PDQ Factorization

Specifically, the PDQ factorization aims to express a matrix $A$ with $m$ rows and $n$ columns as

$$A = PDQ^T,$$  \hspace{1cm} (E)
where $P$ is an $m \times m$ orthogonal matrix, $Q^T$ is the transpose of $Q$, and $D$ is a lower bidiagonal matrix, i.e. whose only nonzero entries are

$$
i = d_{i,i}, \quad i = 1, 2, \ldots, \min(m, n),$$

$$
i = d_{i,i-1}, \quad i = 2, \ldots, \min(m+1, n).$$

Assume $m < n$.

$$
D = \begin{pmatrix}
    d_{11} & 0 & & \\
    d_{12} & d_{22} & & \\
    0 & d_{23} & d_{33} & \\
    & \ddots & \ddots & \ddots \\
    & & d_{m,m-1} & d_{m,m} \\
    & & & \ddots & \ddots \\
    & & & & d_{m,n}
\end{pmatrix}
$$

In our computer program, each column vector of $P$ and $Q$ is a normalized one. Given a (nonzero) initial column (or column vector) $P_1$ from $A$ and, one can construct matrices $P$, $D$, and $Q$ in (E) by some recursive formulae. The important feature of these recursive formulae is that, during the computations, the columns $a_1, a_2, \ldots, a_n$ of $A$ are being referenced sequentially. We also note that, in practice, the inexact arithmetic due to the roundoff error may disturb the orthogonality as well as the normalization. Note that the choice of the initial column $P_1$ affects the whole structure of $P$, $Q$ and $D$. An approximation $A_r$ of $A$ by specific $P$, $Q$ and $D$ is formed as follows:

$$A_r = P_r D_r Q_r^T,$$

where $P_r$, $Q_r$ are the matrices formed from the first $r$ columns of $P$ and $Q$ (in (E)), and $D_r$ is the $r \times r$ lower bidiagonal matrix with $d_{i,i}$, $i = 1, 2, \ldots, r$ on the diagonal and $d_{i,i-1}$, $i = 2, \ldots, r$ on the subdiagonal. Clearly $A_r$ is a reduced matrix (of $A$) of rank $\leq r$.

Moler and Stewart [2] showed, using theoretical studies of the Lanczos algorithm, that $A_r$ will be a fairly accurate approximation to $A$ even for quite small values of $r$. If one compares the storage for $A$ and $A_r$, their ratio will be
Particularly, when $m = n$, we have

$$R(m, n, r) = \frac{2mr+2r}{n^2}$$

clearly in order to save storage it is necessarily that $2nr < n^2$.

That is

$$2r < n.$$ 

Thus PDQ decomposition will provide significant saving of storage and/or transmitting time if $r$ is chosen much smaller than $n$.

It seems that, in general, for a matrix of low rank (its corresponding image is of relatively low detail), the PDQ factorization will be effective in processing and restoring the picture. In practice, it is not clear how to choose the value of $r$ in order to have an efficient representation of a picture $A_n$ with less computer storage and processing time. However, from our experiments, so far, $\frac{n}{4}$ seems to be a suitable threshold for the reduced rank $r$. Of course, the properties that lead to such low numerical rank remain to be studied.

Recall that [1] Andrews defines a condition number $C_R(G) = d_1/d_k$ in the discussion of the potentially efficient representation of the image (picture) in terms of its eigenimages, where $G$ is the original picture (matrix), $d_1$ is the maximal eigenvalue, and $d_k$ the minimal eigenvalue ($\neq 0$) in the SVD decomposition for $G$.

Similarly, we would like to define in PDQ factorization a condition number for reduced matrix of rank $k$: $C_k(A) = \max \{d_i, \sigma_i, i = 1, 2, \ldots, k\}$, where $d_i = \max \{a_1, a_2, \ldots, a_i\}/\min \{a_1, a_2, \ldots, a_i\}$ and $\sigma_i = \max \{\beta_1, \beta_2, \ldots, \beta_i\}/\min \{\beta_1, \beta_2, \ldots, \beta_i\}$

A threshold for $r$ values would be a value of $k$ such that both $C_k(A)$ and $C_{k+1}(A)$ as well as their difference are large (of course, here the quantity of largeness remains to be decided). Another way that one may measure the closeness between $A$ and $A_r$ is by defining the quantity

$$D(A, A_r) = 1 - \frac{\sum_{i=1}^{r} |a_i| + \sum_{j=1}^{r-1} |\sigma_j|}{\sum_{i=1}^{n} |a_i| + \sum_{j=1}^{n-1} |\sigma_j|}.$$
Clearly if \( r = n \), then \( D(A, A_r) = 0 \). Therefore a threshold that one may consider for the approximated value \( r \) is to require \( D(A, A_r) \) to be sufficiently small. We hope to extend our efforts to include the study of the threshold of \( r \) values in our next report.

III. Description of the Matrix Factorization Software.

Because of the limited memory of the NOVA 800, our principal computing tool, factorization of the image matrix is accomplished by the program MATRIXVAX running on the VAX 11/780. A formatted image disk file 'MATAF' is created by the program VAXDATA on the NOVA 800 and written onto a magnetic tape by the program MAGTAP using a blocking factor equal to 1. Program MATRIXVAX (1) reads the image magnetic tape file into memory on the VAX, (2) factors the image matrix into the matrices \( P, D \) and \( Q^T \) of specified rank \( r \) (3) reconstruct the image \( PDQ^T \) onto a magnetic file and (4) writes the matrices \( P, D \) and \( Q^T \) onto a magnetic tape file.

The image and matrix magnetic tape files created by VAX are written into disk files 'MATAFIN' and 'MATP', respectively, by the program MAGTAP on the NOVA 800. The program WRITMATAFIN writes the reconstructed image stored in 'MATAFIN' on the COMTAL CRT and the program WCOMT writes the COMTAL disk file onto magnetic tape. Images of rank \( r_i \), where \( r_i < r \) are written either on the COMTAL CRT by the program WMTCT or Optronics C-4300 Colorwriter by the Image Writing Software System PWREC.

IV. Test Results (Pictures)

V. PDQ Factorization for Blurred Images

Denote the original image by \( A \) (without loss of generality, we may assume \( A \) is a square matrix). Suppose that the noise is modeled by the presence of two unknown matrices \( N_1 \) and \( N_2 \) such that for any given image \( A \), \( B = N_1 A + N_2 \) will be the blurred image of \( A \). In order to apply PDQ factorization to \( A \), we first have to determine the two unknown matrices. To do this we use identity matrices \( I \) and \( 2I \) to define

\[
B_1 = N_1 I + N_2 \quad \text{and} \quad B_2 = 2N_1 I + N_2.
\]

Then \( B_1 \) and \( B_2 \) are the two known matrices. Therefore \( N_1 = B_2 - B_1 \) and \( N_2 = 2B_1 - B_2 \). Once \( N_1 \) and \( N_2 \) are obtained and stored, then for each blurred image \( B \), we apply \( N_1^2 (B - N_2) \) (provided that \( N_1^{-1} \) exists). We get the original image \( A \) and then proceed with PDQ factorization.
VI. Concluding Remarks

The above results will be used for further exploration of the PDQ factorization to digital images. As part of this work efficiency of factorization needs to be improved.

So far, we have seen, through some tests that the PDQ factorization did provide significant saving of storage as well as processing time in the representation of some ordinary pictures. We hope that further research will shed some light on the underlying theory. Our next topics of studies will be (i) properties of images (matrices) whose PDQ factorizations produce good (effective) approximations (i.e., the effective numerically reduced rank r is much less than the rank n of the original matrix, say, rank r less than n/4). (ii) Applying PDQ factorization to the enhancement of the images.

VII. List of Programs

```
C PROGRAM NAME: VAXDATA
C
C INPUT DATA TO VAX FROM TAPE
C
PARAMETER N=128,N=256
DIMENSION IA(N),FB(N),IB(N)
COMMON/PPAP/A,B
DATA A,B/9.,1./
FF(1)=A+B
ACCEPT 'MATRIX DIMENSION = ',NA
ACCEPT 'MATRIX RANK = ',NR
CALL OPEN(2,'MATAF',2,IER)
WHITE(2,1B)NA,NR
10 FORMAT(1X,2(I3,1X))
ACCEPT 'TAPE FILE NUMBER = ',NFILE
IF(NFILE.EQ.0)GO TO 100
CALL SKFMT(IER,NFILE)
100 IL=0
CALL HEADM(TIER,IA,IIL)
IF(IEM.NE.3)GO TO 100
110 DU 130 J=1,NA
120 CONTINUE
GO TO 125
125 DU 20 I=1,12
K=(I-1)*8+1
K2=K+7
WHITE(2,3U)(IB(L),L=K1,K2)
30 FORMAT(1X,8(I6,2X))
20 CONTINUE
ICNT=ICNT+1
IF(ICNT.EQ.NA)GO TO 150
IF(IEN.EQ.5)GO TO 110
GO TO 1WP
150 CONTINUE
DU 5F J=1,NA
P(1)=FF(FLOAT(J))
```

```
CONTINUE
DO 48 L=1,12
K=0.001*N(10) + 1
K2=K1+7
CALL CLDSCRENE
CALL CLDSCA
34 TOAT

HATEX VAX
18147148
CURRENT DIR, YANG

MATRXXAX

88/20/81
10147148

C HDOSFILE NAP4 MATRXXAX
C MATRIX FACTORIZATION
PARAMETER N=10, M=10
COMMON/MATH/ G(N,N), P(N,N), PT(N), QT(N), PHI(N), PHI(N), T(E(N),N),
ALPHA(N), BETA(N), IA(N,N), IA(N,N), IA(N,N)

OPEN(UNIT=21, TYPE='OLD', FILE='MATR.DAT')
OPEN(UNIT=22, TYPE='NEW', FILE='MATR.DAT')
READ(21,400) NA, NR
400 FORMAT(I16,2X)
        CONTINUE
        DO 40 I=1,20
        READ(21,430) (A(L,J), L=K1,K2)
        430 FORMAT(I16,2X)
        410 CONTINUE
        420 CONTINUE
        DO 44 I=1,32
        K1=0.001*N(10) + 1
        K2=K1+7
        READ(21,450) (P(L,1), L=K1,K2)
        450 FORMAT(I16,2X)
        440 CONTINUE
        BETA(1)= 0.
        SUM= 0.
        DO 2 I=1,NA
        SUM=SUM+P(I,1)*P(I,1)
        2 CONTINUE
        SUM=SUM(T(SUM)
        DO 4 I=1,NA
        P(I,1)=P(I,1)/SUM
        4 CONTINUE
        DO 10 K=1,NA
        SUM=SUM+FLOAT(IA(K,1))*P(K,1)
        10 CONTINUE
        G(I,1)=SUM
        29 CONTINUE
        SUM= 0.
        DO 30 I=1,NA
        SUM=SUM+G(I,1)*G(I,1)
        30 CONTINUE
        ALPH(A(1)=SUM(T(SUM)
        37 CONTINUE
        SUM=SUM+G(I,1)*ALPHA(I)
        38 CONTINUE
        STOP
CONTINUE

J=1
IF(J,GT,NH)GO TO 230
DO 79 J2=1,NA
SUM=0
DO 66 K=1,NA
SUM=SUM+FLOAT(JA(J2,K))*Q(K,J=1)
CONTINUE

PT(J2)*SUM=ALPHA(J=1)*P(J2,J=1)
79 CONTINUE
K2=J=1
DO 190 J2=1,NA
SUM=SUM+PT(JJ)*P(JJ,K)
CONTINUE
SUM1=SUM1+SUM=P(J2,K)
90 CONTINUE
PT1(J2)=PT(J2)=SUM1
100 CONTINUE
SUM=0
DO 110 I=1,NA
SUM=SUM+PT1(I)=PT1(I)
CONTINUE
BETA(J)=SRT(SUM)
DO 120 I=1,NA
P(I,J)*PT1(I)/BETA(J)
120 CONTINUE
DO 170 J2=1,NA
SUM=0
DO 160 K=1,NA
SUM=SUM+FLOAT(JA(K,J2))*P(K,J)
160 CONTINUE
QT(J2)*SUM=BETA(J)*Q(J2,J=1)
170 CONTINUE
K2=J=1
DO 290 J2=1,NA
SUM=0
DO 190 K=1,K2
SUM=0
DO 160 JJ=1,NA
SUM=SUM+QT(JJ)*Q(JJ,K)
180 CONTINUE
SUM1=SUM1+SUM=Q(J2,K)
190 CONTINUE
QT1(J2)=QT(J2)=SUM1
200 CONTINUE
DO 210 I=1,NA
SUM=SUM+QT1(I)=QT1(I)
210 CONTINUE
ALPHA(J)=SRT(SUM)
DO 220 I=1,NA
Q(I,J)*QT1(I)/ALPHA(J)
220 CONTINUE
GO TO 50
230 CONTINUE
DO 330 K=1,NA
DO 320 I=1,NA
SUM=0
DO 310 J=1,MP
SUM=SUM+P(I,J)+ALPHA(J)*Q(K,J)
IF(J,LE,1)GO TO 310
SUM=SUM+P(I,J)*BETA(J)*Q(K,J=1)
310 CONTINUE
IAR(I,K)=IFIX(SUM)
320 CONTINUE
DO 330 J=1,200
DO 330 I=1,32
  K1=(I-1)*8+1
  K2=K1+7
  WRITE(23,431) (IAR(L,J1),L=K1,K2)
  CONTINUE

  FORMAT(1X,8(I9,2X))

  X ACCEPT JYN
  READ(11,348) JYN
  CONTINUE(CONTINUE? (I=YES,0=NO) 'I',I3)

  IF(JYN,400,488)
  CONTINUE

  X ACCEPT JYN
  READ(11,358) NR1
  CONTINUE

  X FORMAT('MATRlX RANK = 1',I3)

  CONTINUE

  X IF(JYN,460,400)

  CONTINUE

  X WRITE(24,368) (ALPHA(L),L=K1,K2)
  WRITE(24,369) (BETA(LL),LL=K1,K2)
  CONTINUE

  X WRITE(1X,4(E12,5.1X))

  CONTINUE

  X WRITE(1X,4(E12,5.1X))

  IF(JYN,500,590)

  CONTINUE

  X CLOSE(Unit=21,DISPOSE='DELETE')
  X CLOSE(Unit=23,DISPOSE='SAVE')
  X CLOSE(Unit=24,DISPOSE='SAVE')
  STOP
  END

******* WRMTAFAIN *********************** WRMTAFAIN

08/28/61 18146117
CURRENT DIR. YANG

***********************************************************************
READ(1,38)(IA(L),L=K1,K2)
WRITE(1d,38)(IA(LL),LL=K1,K2)
FORMAT(I1,8(16,2X))
DO 40 M=K1,K2
   IF(IA(M).LT.0)IA(M)=0
   IF(IA(M).GT.255)IA(M)=255
40 CONTINUE
CONTINUE
CALL PACK(IA,IB,NA)
CALL MTIMAGE(IER,IOUT,J,IB,NA/2)
CONTINUE
CALL CLOSE(1,IER)
STOP
END

************************************************************** MATPOTP

08/28/81   10:48:13
CURRENT DIR, YANG

**************************************************************

C
C HOUSE FILE NAME=MTAPOTP
C
C IMAGE CONVERSION BY MULTIPLICATION OF P,Q AND O
C
C MATRIX RANK NB RUN ON YAX MUST BE MULTIPLE OF 4
C PARAMETER N=1284,M=2846,NN=256,NN=128
C DIMENSION P(N),Q(N),ALPHA(NN),SETA(NN),IA(NN),IB(NN)
COMMON/INTS1/IP(M)
COMMON/INTS2/IO(M)
EQUIVALENCE (IP(1),P(1)),(IQ(1),Q(1))
ACCEPT 'NEW RUN?' (YES,NO) = 'Y',JYN
ACCEPT 'MAX MATRIX RANK NB = ',MR
ACCEPT 'TO MTOUT',IFL
CALL OPEN(3,'MATP',2,IER)
NLINES=1224/NN8
NNEW=NN/4
NLS=1224
NLS=8
IF(4=MR/4)GO TO 12
K1=I=1,12
K2=I=3
READ(3,24)(ALPHA(L),L=K1,K2)
READ(3,24)(SETA(L),L=K1,K2)
24 FORMAT(I1,4(E13.8,1X))
CONTINUE
IF(JYN.EQ.0)GO TO 32
CALL CFILH('PHAT',3,208,IER)
IF(IER.NE.1)TYPE 'FILE ERROR'
CALL CFILH('OMAT',3,208,IER)
IF(IER.NE.1)TYPE 'FILE ERROR'
32 CALL OPEN(1,'PHAT',2,IER)
CALL OPEN(2,'OMAT',2,IER)
IF(JYN.EQ.0)GO TO 92
IMAX=NNEW/4
DO 59 J=1,NTRANS
DO 58 I=1,IMAX
K1=I=1,12
K2=I=3
READ(3,28)(P(L),L=K1,K2)
READ(3,28)(Q(LL),LL=K1,K2)
CONTINUE
CALL HSL1A(1,J-1)=NBLKS,IP,NBLKS,IER)
CALL HSL1K(2,J-1)=NBLKS,IP,NBLKS,IER)
X TYPE 'IP(I),I=1,32',(IP(IXXX),XXX=1,32)
X TYPE 'IQ(I),I=1,32',(IQ(IXXX),XXX=1,32)
CONTINUE
50 CALL RETURN('HT',IER)
CONTINUE
52 CALL INIT('HT',IER)
IF(IFL.EQ.0)GO TO 200
CALL SAFLNTIER,IER)
CONTINUE
200 DO 160 K=1,MP
160 CALL ROBK(2,1*NBLKS,NBLKS,IER)
CONTINUE
DO 160 K=1,NLINES
KJ=(K-1)*NMP
DO 160 N=1,NLINES
I=(I-1)*NMP
SUM=P
SUM=SUM+PIJ*ALPHA(J)+U(KJ*J)
IF(J,J.EQ.1)GO TO 160
CONTINUE
160 CALL RETURN('SUM')
CONTINUE
140 IA(II)=II
CONTINUE
141 IF(IA(I),LT,0)IA(II)=0
IF(IA(II),GT,255)IA(II)=255
CONTINUE
150 CALL PACK(IA,IB,256)
CONTINUE
160 CALL TEMPDFIER)
CONTINUE
170 CALL CLOCK(1,IER)
CONTINUE
180 CALL CLOCK(3,IER)
CONTINUE
190 CALL NLSE('NT6',IER)
STOP
END

Acknowledgement

We would like to thank Dr. Warren W. Willman for his assistance in running the program MATXVAX on VAX 11/780. Also thanks to Dr. Igor Jurkevich for his helpful suggestions as to the form of this report.

References

