

```

SUBROUTINE CAMEROT
MAKE UP CAMERA ROTATION MATRIX

CDE PACKAGE
* * NOTE * * DIMENSION DECLARATION FOR Z IS REMOVED, * *
000002 COMMON/COM3DP/XMIN,XMAX,YMIN,YMAX,ZMIN,ZMAX,ZLIM(2),PLOTX,PLOT
1PLOTORG(2),CAMXYZ(3),MX,NY,FMX,FNY,KBEAM,ZETA,CAMWKG(32),XORG(
2GX(3),FX(2),KSCALE,ZORG,NOUT,CENTER(2),PQLIMIT
000002 DIMENSION LIMIT(2),FLIM(2),AMTX(3,3),KCAMWKG(32)
000002 OEQUIVALENCE(U,CAMXYZ),(V,CAMXYZ(2)),(W,CAMXYZ(3)),(MX,LIMIT),
1(FMX,FLIM),(AMTX,CAMWKG(17)),(CAMWKG,KCAMWKG)
END CDE

LAYOUT OF CAMWKG IS SAME AS CAMDT IN RTN CAMERAS

LOCAL CDE
000002 DIMENSION AU(3),AV(3),AW(3)
HANDEDNESS PARAMETER, =1 FOR LEFT-HANDED USUALLY
000002 DATA EPSILON/=1.0/

S=0.0
000003 DO1J=1,3
000005 AV(J)=0.0
000006 AW(J)=0.0
000007 AU(J)=CAMWKG(J+8)-CAMWKG(J)
000012 1 S=S+AU(J)**2
000015 S=SORT(S)
000020 DO2J=1,3
000021 2 AU(J)=AU(J)/S
000025 SIGMA=SORT(AU(1)**2+AU(2)**2)
PREPARE LOOKING STRAIGHT UP OR DOWN
000032 AV(1)=1.0
000033 AW(2)=-EPSILON
000035 IF(AU(3).GT.0.0)AW(2)=-AW(2)
000037 IF(SIGMA.LT.1.0E-3)GO TO 4
X AXIS
000042 AV(1)=AU(2)/SIGMA
000043 AV(2)=-AU(1)/SIGMA
000045 AV(3)=0.0
Y AXIS
000046 AW(1)=EPSILON*AU(1)*AU(3)/SIGMA
000051 AW(2)=EPSILON*AU(2)*AU(3)/SIGMA
000053 AW(3)=-EPSILON*SIGMA
TRANSFER AXIS DIRECTION COSINES TO ROTATIO MATRIX ROWS
000055 4 DO3J=1,3
000057 AMTX(1,J)=AV(J)
000062 AMTX(2,J)=AW(J)
000065 3 AMTX(3,J)=AU(J)
000071 OPRINT 500,CAMWKG(1),CAMWKG(2),CAMWKG(3),CAMWKG(9),CAMWKG(10),
1CAMWKG(11),((AMTX(J,K),K=1,3),J=1,3)
000123 RETURN

500 OFORMAT(*OCAMERA DATA*/*OLENS LOCATION*,3F14.5/*OAIMING POINT*,
1 3F14.5/*OROTATION MATRIX*//((3E15,6))
000124 END
*FOLLOWING VARIABLES EQUIVALENCED BUT NOT REFERENCED
V

```

W
U
MX
FMX

RUN VERSION 2,3 - CDC LEVEL 26 - NYU LEVEL 11

CAMEROT

SUBPROGRAM LENGTH
000200

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS

2 - 000021 4 - 000055 500 - 000132

BLOCK NAMES AND LENGTHS
COM3DP - 000103

VARIABLE ASSIGNMENTS

AMTX	-	000045C01	AU	-	000162	AV	-	000165	AW	-	000170
CAMWKG	-	000025C01	CAMXYZ	-	000014C01	CENTER	-	000100C01	EPSILON	-	000173
FLIM	-	000021C01	FX	-	000073C01	GX	-	000070C01	J	-	000175
K	-	000177	KCAMWKG	-	000025C01	LIMIT	-	000017C01	PLOTORG	-	000012C01
S	-	000174	SIGMA	-	000176	XORG	-	000065C01	ZLIM	-	000006C01

START OF CONSTANTS
000126

START OF TEMPORARIES
000145

START OF INDIRECTS
000154

UNUSED COMPILER SPACE
022700

```

SUBROUTINE CHOP4(W,H,XIN,XOUT,INSIDE,NOUT)
000011 DIMENSION XIN(2,2),XOUT(2,6),XC(2,5),XI(2,4),D(4)
000011 DATA XC/10*0,0/
C PREPARE NO CUT
000011 XOUT(1,3)=XIN(1,1)
000012 XOUT(2,3)=XIN(2,1)
000014 XOUT(1,4)=XIN(1,2)
000015 XOUT(2,4)=XIN(2,2)
000017 INSIDE=0 $ NOUT=0
C TEST FOR TOTAL MISS
000020 IF(AMAX1(XIN(1,1),XIN(1,2)),LT,0,0)RETURN
000025 IF(AMIN1(XIN(1,1),XIN(1,2)),GT,W)RETURN
000033 IF(AMAX1(XIN(2,1),XIN(2,2)),LT,0,0)RETURN
000040 IF(AMIN1(XIN(2,1),XIN(2,2)),GT,H)RETURN
000046 NOUTWKG=0
C TEST FOR POINTS INSIDE, FLAG IF OUTSIDE AS THEN APPROPRIATE
C EXTERNAL SEGMENT WILL BE CAUGHT,
C
C CONSIDER POINTS ON EDGE INSIDE IF BOTH ARE IN, ELSE OUTSIDE,
C
C * NOTE * WE MAY GET ZERO-LENGTH SEGMENTS, WE HAVE TO DECIDE WHI
C A POINT ON EDGE, WHEN THERE IS JUST 1 ON EDGE, SHOULD BE CALLE
C IT SHOULD BE IN IF SEGMENT INTERSECTS THE RECTANGLE ELSEWHERE,
C BUT NOT IF IT IS THE ONLY COMMON POINT BETWEEN SEGMENT + RECTAI
C
000047 TX1=XIN(1,1)*(XIN(1,1)=W)
000051 TY1=XIN(2,1)*(XIN(2,1)=H)
000054 TX2=XIN(1,2)*(XIN(1,2)=W)
000056 TY2=XIN(2,2)*(XIN(2,2)=H)
000061 IF(TX1,GT,0,0,OR,TY1,GT,0,0)NOUTWKG=1
000071 IF(TX2,GT,0,0,OR,TY2,GT,0,0)NOUTWKG=NOUTWKG+2
C
C IF BOTH INSIDE, NOUTWKG WILL BE ZERO
000102 IF(NOUTWKG,EQ,0)GO TO 20
C NOT BOTH INSIDE, FILL IN CORNERS + LOOK FOR INTERSECTIONS,
000103 XC(1,2)=W $ XC(1,3)=W
000105 XC(2,3)=H $ XC(2,4)=H
000107 N=1
000110 DO6J=1,4
C MIRBANE VERSION OF 13JAN69
000111 IF(MIRBANE(XOUT(1,3),XOUT(1,4),XC(1,J),XC(1,J+1),XI(1,N)))11,6
C INTERSECTION, RECORD DISTANCE FROM X1
000135 7 D(N)=(XIN(1,1)-XI(1,N))**2+(XIN(2,1)-XI(2,N))**2
000147 N=N+1 $ INSIDE=2 $ NOUT=NOUTWKG
000153 6 CONTINUE
000155 IF(N=2)11,10,15
C ONLY ONE INTERSECTION, NOUTWKG GIVES CONFIGURATION,
000160 10 IF(NOUTWKG,EQ,2)GO TO 12
C POINT 1 IS OUTSIDE, POINT 2 INSIDE
000162 XOUT(1,1)=XI(1,1)
000163 XOUT(2,1)=XI(2,1)
000165 XOUT(1,2)=XIN(1,2)
000166 XOUT(2,2)=XIN(2,2)
000170 XOUT(1,4)=XI(1,1)
000171 XOUT(2,4)=XI(2,1)
000173 11 RETURN

```

```

C      POINT 2 IS OUTSIDE, POINT 1 INSIDE
000174      12      XOUT(1,1)=XIN(1,1)
000175      XOUT(2,1)=XIN(2,1)
000177      XOUT(1,2)=XI(1,1)
000200      XOUT(2,2)=XI(2,1)
000202      XOUT(1,5)=XI(1,1)
000203      XOUT(2,5)=XI(2,1)
000205      XOUT(1,6)=XIN(1,2)
000206      XOUT(2,6)=XIN(2,2)
000210      RETURN
C      TWO INTERSECTIONS, SORT OUT, FIND CLOSEST TO X1.
000210      15      J=1
000211      IF(D(2),LT,D(1))J=2 $ K=3-J
000217      XOUT(1,1)=XI(1,J)
000221      XOUT(2,1)=XI(2,J)
000223      XOUT(1,2)=XI(1,K)
000225      XOUT(2,2)=XI(2,K)
000227      XOUT(1,4)=XI(1,J)
000231      XOUT(2,4)=XI(2,J)
000233      XOUT(1,5)=XI(1,K)
000235      XOUT(2,5)=XI(2,K)
000237      XOUT(1,6)=XIN(1,2)
000241      XOUT(2,6)=XIN(2,2)
000242      RETURN
C      TOTALLY INSIDE
000243      20      NOUT=0 $ INSIDE=2
000245      XOUT(1,1)=XIN(1,1)
000246      XOUT(2,1)=XIN(2,1)
000247      XOUT(1,2)=XIN(1,2)
000251      XOUT(2,2)=XIN(2,2)
000252      RETURN
000253      END

```

RUN VERSION 2,3 - CDC LEVEL 26 - NYU LEVEL 11

CHOR4

SUBPROGRAM LENGTH

000356

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS

6	=	000153	7	=	000135	10	=	000160	11	=	000173
12	=	000174	15	=	000210	20	=	000243			

BLOCK NAMES AND LENGTHS

VARIABLE ASSIGNMENTS

D	=	000342	J	=	000354	K	=	000355	N	=	000353
NOUTWKG	=	000346	TX1	=	000347	TX2	=	000351	TY1	=	000350
TY2	=	000352	XC	=	000320	XI	=	000332			

START OF CONSTANTS

000255

START OF TEMPORARIES

000256

START OF INDIRECTS

000474

UNUSED COMPILER SPACE

022400

SUBROUTINE DRAWBOX(X,Y,W,H,KBEAM)

ROUTINE TO DRAW BOX, LOWER LEFT CORNER AT (X,Y), WIDTH W (I.E.
X DIRECTION), HEIGHT H, CRT SPOT INTENSITY KBEAM,
BEAM LEFT POSITIONED AT POINT X,Y.

```
000010 CALL CALCMP(X,Y,0,1)
000012 CALL CALCMP(X,Y+H,KBEAM,1)
000021 CALL CALCMP(X+W,Y+H,KBEAM,1)
000032 CALL CALCMP(X+W,Y,KBEAM,1)
000041 CALL CALCMP(X,Y,KBEAM,1)
000046 RETURN
000047 END
```

RUN VERSION 2,3 - CDC LEVEL 26 - NYU LEVEL 11

DRAWBOX

SUBPROGRAM LENGTH

000060

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS

BLOCK NAMES AND LENGTHS

VARIABLE ASSIGNMENTS

START OF CONSTANTS

000051

START OF TEMPORARIES

000054

START OF INDIRECTS

000060

UNUSED COMPILER SPACE

023300

```

SUBROUTINE DRAWPQ(Z)
DRAW VISIBLE PART OF SEGMENT PQ-QC
C
C
C
C
CDE PACKAGE
  *** NOTE *** Z DIMENSION DECLARATION IS REMOVED ***
000003 0COMMON/COM3DP/XMIN,XMAX,YMIN,YMAX,ZMIN,ZMAX,ZLIM(2),PLOTX,PLOT
1PLOTORG(2),CAMXYZ(3),MX,NY,FMX,FNY,KBEAM,ZETA,CAMWKG(32),XORG(
2GX(3),FX(2),KSCALE,ZORG,NOUT,CENTER(2),PQLIMIT
000003  DIMENSION LIMIT(2),FLIM(2),AMTX(3,3),KCAMWKG(32)
000003 0EQUIVALENCE(U,CAMXYZ),(V,CAMXYZ(2)),(W,CAMXYZ(3)),(MX,LIMIT),
1(FMX,FLIM),(AMTX,CAMWKG(17)),(CAMWKG,KCAMWKG)
C
C
C
CDE PACKAGE FOR DRAW3D, DRAWPQ, GNOMON
000003 0COMMON/COM3DPA/PC(3),QC(3),P(3),Q(3),ENDA(6),ENDB(6),OLDQ(4),
1PW(3),QW(3),T(6),PK(3),QK(3),PHIP,PHIQ,PHIA,IBEAM,KDIM
000003  INTEGER PHIP,PHIQ,PHIA
C
C
C
CDE PACKAGE
000003  P(1)=PC(1)
000005  P(2)=PC(2)
000006  P(3)=PC(3)
000010  Q(1)=QC(1)
000011  Q(2)=QC(2)
000013  Q(3)=QC(3)
C
C
000014 2  TEST IF P VISIBLE
IF(PHIP,EQ,0)GO TO 30
C
000015 7  YES, TEST Q
IF(PHIP*PHIQ)10,4,3
C
000020 3  BOTH VISIBLE * SEGMENT DRAWABLE, PLOT + EXIT
KGOTO=0
000021  GO TO 300
C
000022 4  Q IS INVISIBLE, FIND LAST VISIBLE POINT ON SEGMENT PQ
JGOTO=1
000023  GO TO 200
C
000024 5  GIVE UP IF NOT FOUND IN MAXCUT1 BISECTIONS
IF(KFLAG,NE,0)GO TO 6
C
000025  IBEAM=0
000026  GO TO 50
C
000027 6  POINT FOUND
Q(1)=ENDA(1)
000031  Q(2)=ENDA(2)
000032  Q(3)=ENDA(3)
000034  GO TO 3
C
C
000034 10  GAP IN SEGMENT, FIND LAST POINT TO CONNECT TO P,
JGOTO=2
000035  GO TO 200
C
000036 11  IF NOT FOUND (CANNOT FIND POINT WITH SAME VISIBILITY FN), TRY
IF(KFLAG,EQ,0)GO TO 15
C
000037  SAVE OLD Q, RESET TO END POINT + PLOT THIS PIECE,
OLDQ(1)=Q(1)
000041  OLDQ(2)=Q(2)
000042  OLDQ(3)=Q(3)
000044  Q(1)=ENDA(1)
000045  Q(2)=ENDA(2)

```

```

000046      Q(3)=ENDA(3)
C          DRAW FIRST PART OF SEGMENT AND COME BACK HERE
000047      KGOTO=2
000050      GO TO 300
C          RESTORE Q + FIND LOWER LIMIT OF UPPER SEGMENT,
C          LIMITS FOR SEARCH
000050      12      P(1)=Q(1)
000052      P(2)=Q(2)
000053      P(3)=Q(3)
000055      Q(1)=OLDQ(1)
000056      Q(2)=OLDQ(2)
000057      Q(3)=OLDQ(3)
C          BEAM OFF FIRST
000060      15      IBEAM=0
000061      JGOTO=3
000062      GO TO 201
C          IF SEGMENT TOO SHORT, GIVE UP,
000063      13      IF(KFLAG,EQ,0)GO TO 50
C          LOWER END NOW NEWLY FOUND POINT,
000064      14      P(1)=ENDA(1)
000066      P(2)=ENDA(2)
000067      P(3)=ENDA(3)
000071      GO TO 3
C          P INVISIBLE, CHECK Q, IF INVISIBLE, ADVANCE,
000071      30      IBEAM=0
000072      IF(PHIQ,EQ,0)GO TO 50
C          FIND P
000073      JGOTO=4
000074      GO TO 201
C          IF NO POINT, GIVE UP,
000075      31      IF(KFLAG,EQ,0)50,14
C
C
C          P VISIBLE, Q INVISIBLE, FIND Q,
C          ENDB = INVISIBLE END OF INTERVAL, ENDA = VISIBLE
000101      200      ENDB(1)=Q(1)
000103      ENDB(2)=Q(2)
000104      ENDB(3)=Q(3)
000106      ENDA(1)=P(1)
000107      ENDA(2)=P(2)
000111      ENDA(3)=P(3)
C          REQUIRED IVIS FUNCTION
C          IN CASE OF GAP IN SEGMENT, CONSIDER POINT VISIBLE IF ITS VISIB
C          FUNCTION MATCHES THIS ONE AND UPDATE ENDA, ELSE ENDB,
000112      PHIA=PHIP
000114      GO TO 205
C          P INVISIBLE, Q VISIBLE, FIND P,
000114      201      ENDB(1)=P(1)
000116      ENDB(2)=P(2)
000117      ENDB(3)=P(3)
000121      ENDA(1)=Q(1)
000122      ENDA(2)=Q(2)
000124      ENDA(3)=Q(3)
000125      PHIA=PHIQ
000127      205      KFLAG=0
C          GET PROJECTED LENGTH OF SEGMENT
000130      PK(1)=XMIN+(ENDA(1)-1.0)*GX(1)-CAMWKG(1)
000136      PK(2)=YMIN+(ENDA(2)-1.0)*GX(2)-CAMWKG(2)

```

```

000143 PK(3)=ENDA(3)*GX(3)+ZORG=CAMWKG(3)
000147 CALL ROTATE(PK,AMTX,ENDA(4))
000152 PK(1)=XMIN+(ENDB(1)-1,0)*GX(1)-CAMWKG(1)
000160 PK(2)=YMIN+(ENDB(2)-1,0)*GX(2)-CAMWKG(2)
000165 PK(3)=ENDB(3)*GX(3)+ZORG=CAMWKG(3)
000171 CALL ROTATE(PK,AMTX,ENDB(4))
C
000174 210 NEXT STEP
000177 T(1)=(ENDA(1)+ENDB(1))/2,0
000202 T(2)=(ENDA(2)+ENDB(2))/2,0
000205 T(3)=(ENDA(3)+ENDB(3))/2,0
000207 T(4)=(ENDA(4)+ENDB(4))/2,0
000212 T(5)=(ENDA(5)+ENDB(5))/2,0
000215 T(6)=(ENDA(6)+ENDB(6))/2,0
000225 MFLAG=IVIS(T(1),T(2),T(3),Z,KDIM)
IF(MFLAG,EQ,PHIA)GO TO 220
C
000227 NOT VISIBLE, RESET INVISIBLE END
000230 ENDB(1)=T(1)
000232 ENDB(2)=T(2)
000233 ENDB(3)=T(3)
000235 ENDB(4)=T(4)
000236 ENDB(5)=T(5)
000236 ENDB(6)=T(6)
C
000240 216 CHECK SEGMENT LENGTH (USE MAX OF X, Y DIFFERENCES)
OSL=CAMWKG(4)*AMAX1(ABS(ENDA(4)/ENDA(6)-ENDB(4)/ENDB(6)),
1 ABS(ENDA(5)/ENDA(6)-ENDB(5)/ENDB(6)))
000255 IF(SL,GE,PQLIMIT)GO TO 210
000257 GO TO(5,11,13,31),JGOTO
C
000267 220 RECORD VISIBLE, UPDATE ENDA
000271 KFLAG=MFLAG
000272 ENDA(1)=T(1)
000274 ENDA(2)=T(2)
000275 ENDA(3)=T(3)
000277 ENDA(4)=T(4)
000300 ENDA(5)=T(5)
000302 ENDA(6)=T(6)
GO TO 216
C
C
C
C
C
000302 300 IF BEAM IS ON, JUST MOVE IT TO Q,
IF(IBEAM,GT,0)GO TO 310
C
000305 MOVE TO P, BEAM OFF,
000312 PK(1)=XMIN+(P(1)-1,0)*GX(1)-CAMWKG(1)
000317 PK(2)=YMIN+(P(2)-1,0)*GX(2)-CAMWKG(2)
000323 PK(3)=P(3)*GX(3)+ZORG=CAMWKG(3)
000325 CALL ROTATE(PK,AMTX,PW)
000333 PW(1)=(PW(1)/PW(3)=XORG(1))*CAMWKG(4)+PLOTORG(1)+CENTER(1)
000341 PW(2)=(PW(2)/PW(3)=XORG(2))*CAMWKG(4)+PLOTORG(2)+CENTER(2)
CALL CALCMP(PW(1),PW(2),0,1)
C
000345 310 MOVE TO Q, BEAM ON, BEAM IS LEFT ON AND AT POINT Q,
000353 QK(1)=XMIN+(Q(1)-1,0)*GX(1)-CAMWKG(1)
000360 QK(2)=YMIN+(Q(2)-1,0)*GX(2)-CAMWKG(2)
000364 QK(3)=Q(3)*GX(3)+ZORG=CAMWKG(3)
000366 CALL ROTATE(QK,AMTX,QW)
000374 QW(1)=(QW(1)/QW(3)=XORG(1))*CAMWKG(4)+PLOTORG(1)+CENTER(1)
000374 QW(2)=(QW(2)/QW(3)=XORG(2))*CAMWKG(4)+PLOTORG(2)+CENTER(2)
000402 CALL CALCMP(QW(1),QW(2),KBEAM,1)

```

000406

IBEAM=KBEAM

000410

IF(KGOTO,NE,0)GO TO 12

C

000412

50

RETURN

000413

END

*FOLLOWING VARIABLES EQUIVALENCED BUT NOT REFERENCED

V

W

U

MX

FMX