

Source: SWEDEN
Title: DISPLAY FACILITIES
Purpose: Information

This document illustrates organisation of facilities for video display.

A special room on 2nd floor is used for video display. In this room we will use a Sony projection monitor, giving a large size picture on the wall. In another room, on 1st floor, the D1 machine, DVS system and control computer is situated. A terminal in the display room is connected to the computer. Continuous contact between the rooms is established with a telephone. See Figure for overview.

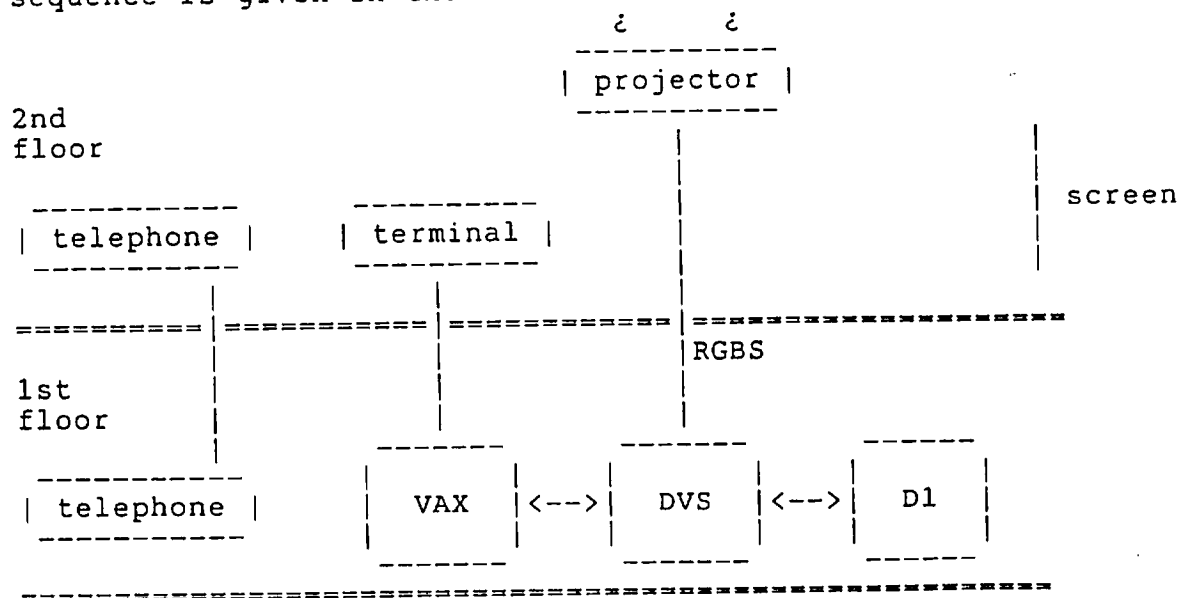
D1 tapes are loaded in the D1 machine by an operator on request on the telephone. The D1 machine is controlled via the terminal by another operator. Direct display is done by specifying time code for start point.

SCIF pictures, sent to TRAB before the meeting, have been converted to internal format and stored on a special D1 tape. They must be loaded to DVS-RAM (takes a few seconds) before display.

A sequence which is loaded into the DVS-RAM can be displayed in several modes. It takes approximately 15 seconds to change display mode. Picture aspect ratio is set by specifying sampling frequency. Some interesting modes are

625/25/int	625/50/int	625/50/pro	625/60/pro (SCIF)
525/30/int		525/60/pro	

In order to check difference between interlace and progressive display, a special syntetic sequence was produced. At the same time, the sequence was designed to visualise any potential weakness with chrominance subsampling. The C-program which generates the sequence is given in the annex.



***** Annex to AVC-225 *****/

```

static char
Y [576][720], U [288][360], V [288][360];
int
i,j,picnr,npic;

for ( picnr=0 ; picnr<npic ; picnr ++ )
{
    /* LIGHT UNCOLORED BACKGROUND */
    /*******/

    for ( j=0 ; j < 576 ; j++ )
        for ( i=0 ; i < 720 ; i++ )
        {
            Y [j][i] = 64 ;
        }

    for ( j=0 ; j < 288 ; j++ )
        for ( i=0 ; i < 360 ; i++ )
        {
            U [j][i] = 0 ;
            V [j][i] = 0 ;
        }

    /* DARK HORIZONTAL LINES MOVING DOWN */
    /*******/
    for ( i=0 ; i < 720 ; i++ )
    {
        Y [ 10+picnr*0 ][i] = -64 ;
        Y [ 20+picnr*1 ][i] = -64 ;
        Y [ 30+picnr*2 ][i] = -64 ;
        Y [ 40+picnr*3 ][i] = -64 ;
        Y [ 50+picnr*4 ][i] = -64 ;
        Y [ 60+picnr*5 ][i] = -64 ;
    }
    for ( i=0 ; i < 360 ; i++ )
    {
        V [10/2+picnr*0/2][i] = 32 ;
        V [20/2+picnr*1/2][i] = 32 ;
        V [30/2+picnr*2/2][i] = 32 ;
        V [40/2+picnr*3/2][i] = 32 ;
        V [50/2+picnr*4/2][i] = 32 ;
        V [60/2+picnr*5/2][i] = 32 ;
    }

    /* DARK VERTICAL LINES MOVING RIGHT */
    /*******/
    for ( j=0 ; j < 576 ; j++ )
    {
        Y [j][ 20+picnr*0 ] = -64 ;
        Y [j][ 40+picnr*1 ] = -64 ;
        Y [j][ 60+picnr*2 ] = -64 ;
        Y [j][ 80+picnr*3 ] = -64 ;
        Y [j][100+picnr*4 ] = -64 ;
        Y [j][120+picnr*5 ] = -64 ;
    }
    for ( j=0 ; j < 288 ; j++ )
    {
        V [j][ ( 20+picnr*0 ) / 2 ] = 32 ;
        V [j][ ( 40+picnr*1 ) / 2 ] = 32 ;
        V [j][ ( 60+picnr*2 ) / 2 ] = 32 ;
        V [j][ ( 80+picnr*3 ) / 2 ] = 32 ;
        V [j][ (100+picnr*4 ) / 2 ] = 32 ;
        V [j][ (120+picnr*5 ) / 2 ] = 32 ;
    }
}

```

```

/* Vertical Y loop */
/* Horizontal Y loop */

/* Light background */

/* Vertical C loop */
/* Horizontal C loop */

/* Uncolored background */

/* Horizontal lines */
/*
/* Motion = none
/* Motion = 1 line/pic
/* Motion = 2 lines/pic
/* Motion = 3 lines/pic
/* Motion = 4 lines/pic
/* Motion = 5 lines/pic
*/

/* Red */

/* Vertical line */
/*
/* Motion = none
/* Motion = 1 pel/pic
/* Motion = 2 pels/pic
/* Motion = 3 pels/pic
/* Motion = 4 pels/pic
/* Motion = 5 pels/pic
*/

/* Red */

```

```

/* GROWING BLUE SQUARES */
/*****/
for ( j= 261+picnr*0 ; j < 449+picnr*5 ; j++ )
  for ( i= 261+picnr*0 ; i < 449+picnr*5 ; i++ )
  {
    Y [j][i] = -64 ; /* Dark */
  }
for ( j= 301+picnr*1 ; j < 419+picnr*4 ; j++ )
  for ( i= 301+picnr*1 ; i < 419+picnr*4 ; i++ )
  {
    Y [j][i] = 64 ; /* Light */
  }
for ( j= 341+picnr*2 ; j < 389+picnr*3 ; j++ )
  for ( i= 341+picnr*2 ; i < 389+picnr*3 ; i++ )
  {
    Y [j][i] = -64 ; /* Dark */
  }

for ( j=(261+picnr*0)/2 ; j < (449+picnr*5)/2 ; j++ )
  for ( i=(261+picnr*0)/2 ; i < (449+picnr*5)/2 ; i++ )
  {
    U [j][i] = 32 ; /* Blue */
  }
for ( j=(301+picnr*1)/2 ; j < (419+picnr*4)/2 ; j++ )
  for ( i=(301+picnr*1)/2 ; i < (419+picnr*4)/2 ; i++ )
  {
    U [j][i] = 0 ; /* No color */
  }
for ( j=(341+picnr*2)/2 ; j < (389+picnr*3)/2 ; j++ )
  for ( i=(341+picnr*2)/2 ; i < (389+picnr*3)/2 ; i++ )
  {
    U [j][i] = 32 ; /* Blue */
  }

/* HORIZONTAL STRIPES */
/*****/
for ( j= 301 ; j < 499 ; j+=2 ) /* Every 2nd line */
  for ( i= 601 ; i < 699 ; i++ )
  {
    Y [j][i] = -64 ; /* Dark */
  }
} /* end of picnr loop */

```