

## Background

Matrox Meteor-II and Matrox Orion<sup>1</sup> include a single video decoder to capture standard video formats (i.e., NTSC/PAL or RS-170/CCIR). However, it is the multiplexer stage that enables one to connect and switch between multiple camera inputs; up to 12 CVBS or 6 Y/C inputs for Matrox Meteor-II<sup>2</sup>, and up to 8 CVBS or 4 Y/C inputs for Matrox Orion. This document explains how to achieve the fastest possible switching speed with a Matrox Meteor-II or Matrox Orion when connected to multiple unsynchronized inputs of the same video format. The topics discussed in this document assumes the use of MIL/MIL-Lite 7, as well as a Matrox Meteor-II with rev. B (or higher) of the video decoder (see Appendix). For the remainder of this document we will use the term frame grabber to designate the Matrox Meteor-II or Matrox Orion<sup>1</sup>.

## Capture Mode

The first step to ensure fast switching is to set the grab mode to synchronous. This is done using `MdigControl(..., M_GRAB_MODE, M_SYNCHRONOUS)`<sup>3,4</sup>. In synchronous mode, the grab command to capture a single frame will return control to the application once the capture operation is performed and not before. Doing so ensures that the capture operation has completed before issuing any other commands such as switching channels. Switching channels before the grab command is completed can prevent the frame grabber's circuitry from being able to determine the proper phase of the video signal (i.e., blanking versus active) quickly enough. This can produce undesired results such as additional delays due to resynchronization, display artifacts such as empty lines (black lines), or split frames (an image with fields from different channels).

The next step is to set the grab start mode to capture on the very next field (even or odd) using `MdigControl(..., M_GRAB_START_MODE, M_FIELD_START)`<sup>5</sup>. This prevents a variable delay during channel switch and the start of the next capture. For example, if the Grab Start Mode is set to odd (`M_FIELD_START_ODD`)<sup>6</sup> and the second channel is currently on an odd field, the frame grabber will wait until the next available odd field before it starts grabbing (see Figure 1).

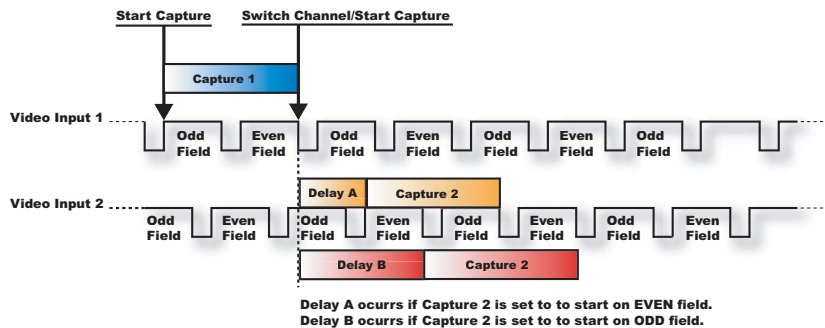


Figure 1: The smallest delay (Delay A) between a channel switch and the start of the next capture will occur when Capture 2 is set to start the very next field (`M_FIELD_START`). When Capture 2 is set to a specific field (such as ODD) a delay (Delay B) as long as two fields can result.

## Channel Switch

Switching channels is performed using `MdigChannel(..., M_CHx)`<sup>7</sup>, where 'x' represents the desired channel. Refer to the MIL/MIL-Lite Board Specific Notes manual for further details related to your specific frame grabber.



## Locking to the Video Source

For fastest channel switching, it is vital to ensure that the frame grabber detects the proper phase of the video signal. Using `MdigControl(..., M_CAMERA_LOCK, M_ENABLE)`<sup>8</sup> ensures that following the channel switch operation the frame grabber is locked to the luminance for monochrome and color video signals, as well as to the chrominance for color video signals.

## Automatic Gain Control (AGC)

Performance can also be improved by reading and saving the AGC value for each channel, disabling the AGC, and then setting the gain to a fixed level following each channel switch to the previously saved value. Start by using `MdigInquire(..., M_GRAB_INPUT_GAIN, ...)`<sup>9</sup> to read and save the gain value for each channel. To disable the AGC, use `MdigControl(..., M_GRAB_AUTOMATIC_INPUT_GAIN, M_DISABLE)`<sup>10</sup>. To specify a gain value use `MdigControl(..., M_GRAB_INPUT_GAIN, x)`<sup>11</sup> where 'x' is an integer value between 0 and 255. Note however that specifying a gain value that is too low can introduce synchronization problems. Some experimentation may be required.

## Other Optimizations

Special versions of the Matrox Meteor-II digitizer configuration format (dcf) files (NTSC\_FS.DCF and PAL\_FS.DCF) provide further optimization by starting capture on the rising edge of the VSYNC rather than the falling edge. However, these are only available for field mode. These dcf's are on the MIL/MIL-Lite CDs (Matrox\Drivers\Meteor-II\DCF).

## Measured Performance

To quantify the actual performance when switching continuously between two video cameras, Matrox Imaging used a capture and switch application that also included a timer to calculate the average combined field rate. Two cameras were connected<sup>12</sup> to a single Matrox Meteor-II/Matrox Orion<sup>1</sup> set to perform a monoshot capture. A channel switch was then performed following the capture of each field. Since the two cameras are unsynchronized and the two video signals drift from each other over time, the results measured are not absolute but rather an average over time. Furthermore the rates achieved will vary depending the board (e.g., two identical Matrox Meteor-II boards can achieve higher or lower rates) as well as camera used. The results are listed in the table below:

Matrox Board	Video Format	Typical Field Rates
Matrox Meteor-II	NTSC (Color)	23.3 fps
Matrox Meteor-II	PAL (Color)	19.9 fps
Matrox Orion for PCI/AGP	NTSC (Color)	16.2 fps
Matrox Orion for PCI/AGP	PAL (Color)	14.4 fps

The average (typical) switching time is equal to one over the typical field rate minus the field time. Thus we can deduce the actual switching time to be:

Matrox Meteor-II for NTSC	$1/23.3 \text{ fps} - 1/60 \text{ fps} = 42.9 \text{ ms} - 16.7 \text{ ms} \gg 26.2 \text{ ms}$
Matrox Meteor-II for PAL	$1/19.9 \text{ fps} - 1/50 \text{ fps} = 50.2 \text{ ms} - 20 \text{ ms} \gg 30.2 \text{ ms}$
Matrox Orion for NTSC	$1/16.2 \text{ fps} - 1/60 \text{ fps} = 61.7 \text{ ms} - 16.7 \text{ ms} \gg 45.0 \text{ ms}$
Matrox Orion for PAL	$1/14.4 \text{ fps} - 1/50 \text{ fps} = 69.4 \text{ ms} - 20 \text{ ms} \gg 49.4 \text{ ms}$



## Example Code

### 1. MIL Example of Fast Switching Between Video Input Channels

```
#include <stdio.h>
#include <mil.h>
#include <conio.h>

#define DCF_NAME      "ntsc_fs.dcf"

#define IMAGE_SCALE 0.5

#define NUM_CHANNELS 2

#define NB_IMAGE_X  2
#define NB_IMAGE_Y  1

void main(void)
{
    MIL_ID MilApplication,
          MilSystem,
          MilDisplay,
          MilDigitizer,
          MilImageDisp,
          MilImageDispChild[NB_IMAGE_X*NB_IMAGE_Y];

    long   Lock;
    long   ColorDigitizer = M_FALSE;
    long   Counter = 0;
    long   NbGrab = 0;
    long   X,Y;
    long   Channel[] = {M_CH0, M_CH1, M_CH2,  M_CH3,
                       M_CH4, M_CH5, M_CH6,  M_CH7,
                       M_CH8, M_CH9, M_CH10, M_CH11};

    long   Color[NUM_CHANNELS] = {0};
    double Wait = 0.25;
    double TimeStart,TimeEnd;
    unsigned long  AGCValue [NUM_CHANNELS];

    MappAllocDefault(M_DEFAULT, &MilApplication, &MilSystem, &MilDisplay, &MilDigitizer,
    &MilImageDisp);
    ,

    MbufClear(MilImageDisp, 128L);

    MdispSelect(MilDisplay, MilImageDisp);

    Counter = 0;
    for (Y = 0; Y < NB_IMAGE_Y; Y++)
    {
        for (X = 0; X < NB_IMAGE_X; X++)
        {

            MbufChild2d(MilImageDisp,
                (long)(MdigInquire(MilDigitizer, M_SIZE_X, M_NULL)*IMAGE_SCALE*X),
                (long)(MdigInquire(MilDigitizer, M_SIZE_Y, M_NULL)*IMAGE_SCALE*Y),
                (long)(MdigInquire(MilDigitizer, M_SIZE_X, M_NULL)*IMAGE_SCALE),
                (long)(MdigInquire(MilDigitizer, M_SIZE_Y, M_NULL)*IMAGE_SCALE),
                &MilImageDispChild[Counter++]);


```

Continued

## Example Code (continued)

```
    }  
  }  
  
  if(MdigInquire(MilDigitizer, M_SIZE_BAND, M_NULL) != 1)  
    ColorDigitizer = M_TRUE;  
  
  MdigControl(MilDigitizer, M_CAMERA_LOCK, M_DISABLE);  
  for (Counter = 0; Counter < NUM_CHANNELS; Counter++)  
  {  
    MdigControl(MilDigitizer, M_GRAB_AUTOMATIC_INPUT_GAIN, M_ENABLE);  
    MdigChannel(MilDigitizer, Channel[Counter]);  
    MappTimer(M_TIMER_WAIT, &Wait);  
    if( ColorDigitizer == M_TRUE)  
    {  
      Lock=M_FALSE;  
      MdigInquire(MilDigitizer, M_CAMERA_LOCKED, &Lock);  
  
      if(Lock == 0)  
        Color[Counter]=0;  
      else  
        Color[Counter]=1;  
    }  
  
    MdigControl(MilDigitizer, M_GRAB_AUTOMATIC_INPUT_GAIN, M_DISABLE);  
    MdigInquire (MilDigitizer, M_GRAB_INPUT_GAIN, &AGCValue[Counter]);  
  }  
  
  MdigControl(MilDigitizer, M_CAMERA_LOCK, M_ENABLE);  
  
  MdigControl(MilDigitizer, M_CAMERA_LOCK_SENSITIVITY, M_DEFAULT);  
  MdigControl(MilDigitizer, M_CAMERA_UNLOCK_SENSITIVITY, M_DEFAULT);  
  
  MdigControl(MilDigitizer, M_GRAB_MODE, M_SYNCHRONOUS);  
  
  MdigControl(MilDigitizer, M_GRAB_START_MODE, M_FIELD_START);  
  
  MdigControl(MilDigitizer, M_GRAB_SCALE, IMAGE_SCALE);  
  
  MdigControl(MilDigitizer, M_GRAB_FIELD_NUM, 1);  
  
  MdigControl(MilDigitizer, M_GRAB_AUTOMATIC_INPUT_GAIN, M_DISABLE);  
  
  printf("Press <Enter> to stop grabbing and exit.\n\n");  
  
  Counter = NUM_CHANNELS -1;  
  
  MappTimer(M_TIMER_RESET, M_NULL);  
  MappTimer(M_TIMER_READ, &TimeStart);  
  while(!kbhit())  
  {  
    if (Counter < NUM_CHANNELS - 1)  
      Counter++;  
    else  
      Counter = 0;  
  
    MdigControl(MilDigitizer, M_GRAB_INPUT_GAIN, AGCValue[Counter]);
```

Continued

## Example Code (continued)

```
    if (Color[Counter])
        MdigControl(MilDigitizer, M_CAMERA_COLOR_LOCK, M_ENABLE);
    else
        MdigControl(MilDigitizer, M_CAMERA_COLOR_LOCK, M_DISABLE);

    MdigChannel(MilDigitizer, Channel[Counter]);
    MdigGrab(MilDigitizer, MilImageDispChild[Counter]);

    NbGrab++;
    if((NbGrab % 100) == 0)
    {
        MappTimer(M_TIMER_READ, &TimeEnd);
        printf("%d fields grabbed at %f fps\r", NbGrab, 100/(TimeEnd-TimeStart));
        MappTimer(M_TIMER_READ, &TimeStart);
    }
}

MappTimer(M_TIMER_READ, &TimeEnd);

printf("\n\n%d Total fields grabbed at an average of %f fps\n", NbGrab,
NbGrab/TimeEnd);
printf("Press <Enter> to end.\n");
getchar();

for (Counter = 0; Counter < NB_IMAGE_X*NB_IMAGE_Y; Counter++)
{
    MbufFree(MilImageDispChild[Counter]);
}

MbufFree(MilImageDisp);
MdigFree(MilDigitizer);
MdispFree(MilDisplay);
MsysFree(MilSystem);
MappFree(MilApplication);
}
```

## Example Code (continued)

### 2. ActiveMIL Example of Channel Switching

```
Option Explicit
```

```
Const IMAGE_SCALE = 0.5
```

```
Const NUMBER_CHANNELS = 2 ' Maximum 12
```

```
Public IsGrabbing As Boolean  
Dim CameraIsColor() As Boolean  
Dim AutomaticInputGainValue() As Long  
Dim Channel(0 To 11) As Long  
Dim ChildImages(0 To NUMBER_CHANNELS - 1) As MIL.IImage  
Dim NbGrab As Long  
Dim TimeEnd As Double  
Dim m_StopGrabLoop As Boolean  
Dim m_KillLoop As Boolean
```

```
Private Sub Form_Load()
```

```
    Channel(0) = digCh0: Channel(1) = digCh1: Channel(2) = digCh2: Channel(3) = digCh3  
    Channel(4) = digCh4: Channel(5) = digCh5: Channel(6) = digCh6: Channel(7) = digCh7  
    Channel(8) = digCh8: Channel(9) = digCh9: Channel(10) = digCh10: Channel(11) =  
    digCh11
```

```
    ReDim CameraIsColor(0 To NUMBER_CHANNELS - 1)  
    ReDim AutomaticInputGainValue(0 To NUMBER_CHANNELS - 1)
```

```
    Digitizer.ScaleX = IMAGE_SCALE  
    Digitizer.ScaleY = IMAGE_SCALE
```

```
    If ImageBuffer.IsAllocated Then ImageBuffer.Free  
    ImageBuffer.NumberOfBands = Digitizer.NumberOfBands  
    ImageBuffer.SizeX = Digitizer.SizeX * IMAGE_SCALE * NUMBER_IMAGE_X  
    ImageBuffer.SizeY = Digitizer.SizeY * IMAGE_SCALE * NUMBER_IMAGE_Y  
    ImageBuffer.Allocate
```

```
    ImageBuffer.Clear 128
```

```
    Dim CurrentChannel As Long  
    Dim y As Long  
    Dim x As Long
```

```
    For y = 0 To NUMBER_IMAGE_Y - 1  
        For x = 0 To NUMBER_IMAGE_X - 1
```

```
            Dim NewChildImage As MIL.IImage  
            Set NewChildImage = Application.CreateObject("MIL.Image", False)
```

```
Continued
```

### Example Code (continued)

```
        With NewChildImage
            .ParentImage = ImageBuffer
            .ChildRegion.OffsetX = Digitizer.SizeX * IMAGE_SCALE * x
            .ChildRegion.OffsetY = Digitizer.SizeY * IMAGE_SCALE * y
            .ChildRegion.SizeX = Digitizer.SizeX * IMAGE_SCALE
            .ChildRegion.SizeY = Digitizer.SizeY * IMAGE_SCALE
            .Allocate

            Set ChildImages(CurrentChannel) = NewChildImage
            CurrentChannel = CurrentChannel + 1
        End With
    Next
Next

AdjustForm Me

Dim ColorDigitizer As Boolean
If Digitizer.NumberOfBands <> 1 Then
    ColorDigitizer = True
End If

Digitizer.CameraLockEnabled = False

For CurrentChannel = 0 To NUMBER_CHANNELS - 1

    Digitizer.AutomaticInputGain = True
    Digitizer.Channel = Channel(CurrentChannel)
    Application.Timer.Wait 0.25

    If ColorDigitizer = True Then
        CameraIsColor(CurrentChannel) = Digitizer.IsCameraLocked
    End If

    Digitizer.AutomaticInputGain = False
    AutomaticInputGainValue(CurrentChannel) = Digitizer.InputGain
Next

Digitizer.CameraLockEnabled = True

Digitizer.CameraLockSensitivity = digDefault
Digitizer.CameraUnlockSensitivity = digDefault

Digitizer.GrabMode = digSynchronous

Digitizer.GrabStartMode = digFieldStart

Digitizer.GrabFieldNumber = 1

Digitizer.AutomaticInputGain = False

m_StopGrabLoop = False
m_KillLoop = False

lblComment = "Click Next to start the grab on two channels. "
```

End Sub

Continued

### Example Code (continued)

```
Private Sub cmdNext_Click()
    AdjustStepNumber Me

    Select Case cmdNext.Tag
        Case 2
            lblComment = "Grab image on all channels. Switch from one " & _
                "channel to other. Click Next to stop grabbing. "

            GrabLoop

        Case 3
            m_StopGrabLoop = True

            TimeEnd = Application.Timer.Read

            lblComment = NbGrab & " Total fields grabbed at an average of " & _
                Format(NbGrab / TimeEnd, "###0.00") & "fps" & _
                ". Click Next to End."

            DoEvents

        Case Else
            Unload Me
    End Select
End Sub

Private Sub GrabLoop()
    Dim TimeStart As Double
    Dim CurrentChannel As Long

    CurrentChannel = NUMBER_CHANNELS - 1

    Application.Timer.Reset
    TimeStart = Application.Timer.Read

    While (Not m_StopGrabLoop And Not m_KillLoop)
        If CurrentChannel < NUMBER_CHANNELS - 1 Then
            CurrentChannel = CurrentChannel + 1
        Else
            CurrentChannel = 0
        End If

        Digitizer.InputGain = AutomaticInputGainValue(CurrentChannel)

        If CameraIsColor(CurrentChannel) Then
            Digitizer.CameraColorLockEnabled = True
        Else
            Digitizer.CameraColorLockEnabled = False
        End If
    End While
End Sub
```

Continued

### Example Code (continued)

```
Digitizer.Channel = Channel(CurrentChannel)
Digitizer.Image = ChildImages(CurrentChannel)

Digitizer.Grab
NbGrab = NbGrab + 1

If (NbGrab Mod 100) = 0 Then
    TimeEnd = Application.Timer.Read
    lblComment = NbGrab & " fields grabbed at " & _
        Format(100 / (TimeEnd - TimeStart), "###0.00") _
        & "fps. Click next to stop grabbing."
    TimeStart = Application.Timer.Read
End If

DoEvents

Wend

End Sub

Private Sub Form_Unload(Cancel As Integer)
    m_KillLoop = True
End Sub
```

# Application Note/ Matrox Meteor-II and Orion

Rapid Switching Between Multiple Video Inputs

## Appendix

To determine which revision of the video decoder is on your Matrox Meteor-II board, simply locate the component on the board as illustrated in Figure 2. A revision 'A' decoder is identified with 'KS0127' as shown. If the decoder is labeled with any other identification, it is a revision 'B'. Note that Matrox Orion boards are only equipped with revision 'B' and higher decoder chips.

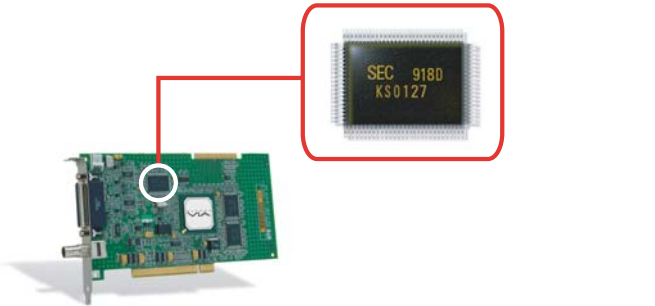


Figure 2: Location of video decoder on Matrox Meteor-II. Revision 'A' of the decoder chip (right) can be easily identified by the 'KS0127' stamp on the chip

### Endnotes:

1. Includes Matrox Orion module for 4Sight-II
2. Up to 7 CVBS or 3 Y/C inputs on CompactPCI® version.
3. Digitizer.GrabMode = digSynchronous; property for ActiveMIL/ActiveMIL-Lite.
4. Synchronous grab mode is the default setting for MIL and ActiveMIL.
5. Digitizer.GrabStartMode = digFieldStart; property for ActiveMIL/ActiveMIL-Lite.
6. Digitizer.GrabStartMode = digFieldStartOdd; property for ActiveMIL/ActiveMIL-Lite.
7. Digitizer.Channel = digChx, where the x represents the desired channel; property for ActiveMIL/ActiveMIL-Lite.
8. Digitizer.CameraLockEnabled = True; property for ActiveMIL/ActiveMIL-Lite.
9. x = Digitizer.InputGain, where the x represents the desired channel; property for ActiveMIL/ActiveMIL-Lite.
10. Digitizer.AutomaticInputGain = False; property for ActiveMIL/ActiveMIL-Lite.
11. Digitizer.InputGain = x, where the x is a integer value between 0 and 255; property for ActiveMIL/ActiveMIL-Lite.
12. In a Dell Dimension 4100 (PIII @ 1 GHz) with a Matrox G400 AGP.

For more information, please call: 1-800-804-6243 (toll free in North America) or (514) 822-6020 or e-mail: [imaging.info@matrox.com](mailto:imaging.info@matrox.com) or <http://www.matrox.com/imaging>

### Corporate headquarters:

Canada and U.S.A.  
Matrox Electronic Systems Ltd.  
1055 St. Regis Blvd.  
Dorval, Quebec H9P 2T4  
Canada  
Tel: (514) 685-2630  
Fax: (514) 822-6273

### Offices:

Europe, Middle East & Africa  
Matrox VITE Limited  
Sefton Park  
Stoke Poges  
Buckinghamshire  
SL2 4JS  
UK  
Tel: 01753 665511  
Fax: 01753 665599

### France

Matrox France SARL  
2, rue de la Couture  
Silic 225  
94528 Rungis Cedex  
Tel: (0) 1 45-60-62-00  
Fax: (0) 1 45-60-62-05

### Germany

Matrox Electronic Systems GmbH  
Inselkammerstr. 8  
D-82008 Unterhaching  
Tel: 089/62170-0  
Fax: 089/614 9743



MATROX  
IMAGING