Matrox Imaging Application Note:

Using the Multiple Matrox Meteor in a System

INTRODUCTION

This document briefly describes how multiple Matrox Meteors can be supported in a single PC system. The use of multiple Matrox Meteor boards can allow a user to grab multiple images simultaneously. It is possible to grab from genlocked cameras or from non-synchronous sources (cameras can be of the same or different formats).

Topics covered below are:

- ✓ installation of the boards;
- ✓ grabbing from genlocked or non-sync cameras;
- ✓ the number of cameras per board;
- ✓ system bandwidth requirements;
- ✓ along with a software example and software support.

MULTIPLE BOARD INSTALLATION

Installation of multiple boards will require an empty PCI (32-bit) slot for each board to be installed, and in some instances, the removal of a PCI card from the system to make room for the additional board. When installing an additional card, it is important to observe the steps followed when installing the first Matrox Meteor board and pay close attention to the existence of a large black-ridged heat sink found in some computers. This heat sink prevents long cards from using most PCI card slots. The Matrox Meteor boards must never come in contact with this heat sink and it is important to use a slot that will allow the boards to completely avoid the heat sink. In the event that the heat sink prevents installation of the additional Matrox Meteors, contact your computer dealer. Through the MIL-Lite software, the user can specify the number of Matrox Meteor boards desired in a system (see SOFTWARE EXAMPLE in this document).

GRABBING FROM GENLOCKED OR NON-SYNC CAMERAS

Grabbing from two genlocked cameras simultaneously allows the capture of two different images at exactly the same time (see figure 1) at different memory locations or different memory buffers. In this case, the cameras must be in the same format. In the case of non-synchronous, the cameras can either be the same format or different formats (see figure 2). The MIL driver version 3.1x supports the display of one live (active window) and one pseudo live image (non-active window) while version 4.0 will support the display of two live images.

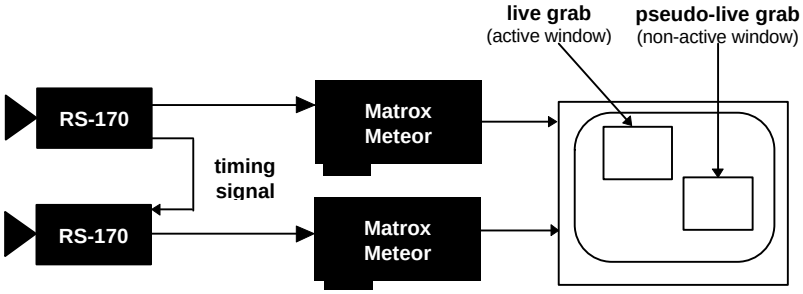

Figure 1: Grabbing from two genlocked cameras

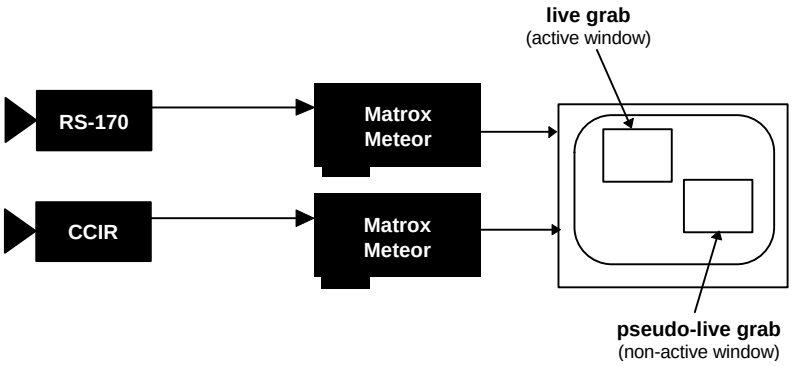

Figure 2: Grabbing from two non-synchronous cameras (with different formats)

HOW MANY CAMERAS PER BOARD?

Since the Matrox Meteor has up to four software selectable video inputs available, a user may connect one or more cameras to each of the Matrox Meteors. However, since only one camera is selected through a multiplexer (see figure 3), only one image will be grabbed from one camera per board at a time. The image from each one of the cameras will be grabbed and displayed as a live (active window) or pseudo live (non-active window).

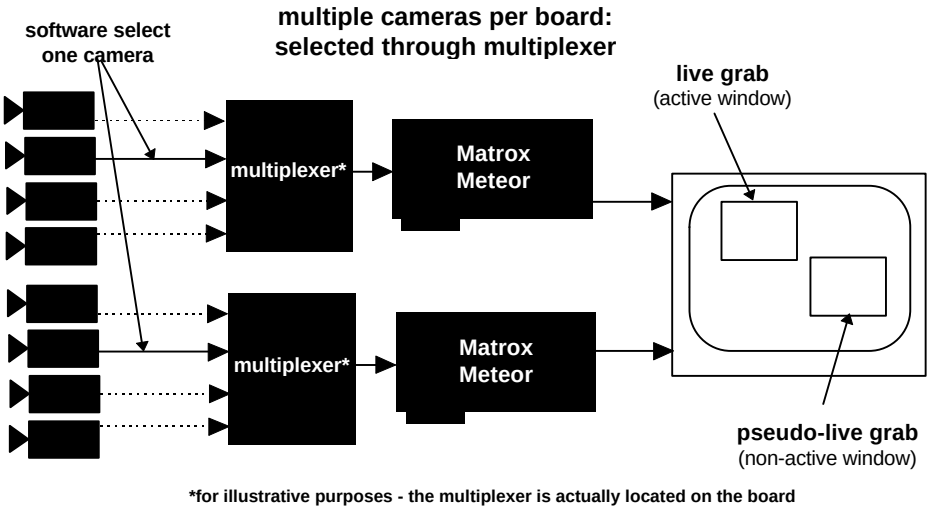

Figure 3: Grabbing from multiple cameras connected to each board

PCI BANDWIDTH REQUIREMENTS

Sustained PCI transfers to memory require the use of a high performance PCI core logic chipset, such as the Intel 430 FX (Triton). If a high performance chipset is used, using two Matrox Meteors in a system for simultaneously grabbing two full size images at full rate should not present any PCI bandwidth problems. However, using more than two Matrox Meteors for simultaneously grabbing more than two images may result in PCI bandwidth problems.

For example, grabbing one full sized NTSC or PAL image at full rate will require a PCI bandwidth of 35 MB/second and 42 MB/second respectively.

NTSC (full-sized)= 640 x 480 at 32-bits/pixel at a rate of 30 frames per second ≈ 35 MB/second

PAL (full-sized)= 768 x 576 at 32-bits/pixel at a rate of 25 frames per second ≈ 42 MB/second

When grabbing from three or more Matrox Meteor simultaneously, reduction of the image size will be necessary in order to avoid reaching the upper limits of overall bandwidth.

WINDOW OCCLUSION

The Matrox Meteor does not support the occlusion of two live windows when grabbing two images directly to display. Since the Matrox Meteor has no on-board display buffer, it uses the VGA card's buffer memory. With MIL driver version 3.1x, any overlap of windows on the display device will result in one live and one pseudo live image at 15 frames per second (see figure 4). With the upcoming MIL driver 4.0, performance will be improved to 27 frames per second. An improved detection of window overlap will also exist; for a simple overlap there will be live grab capability for both windows (see figure 5).

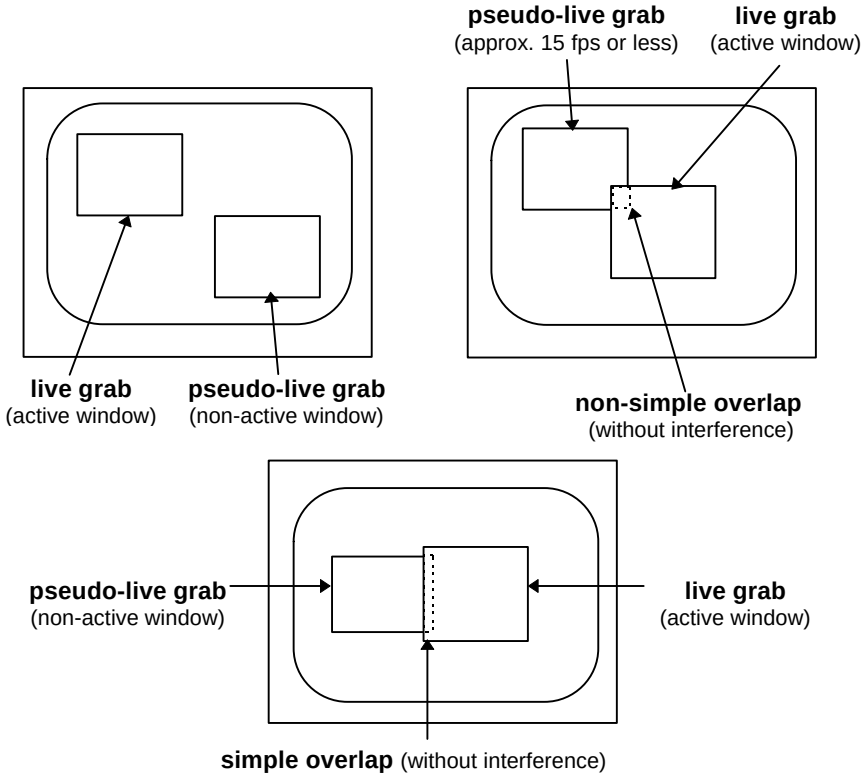

Figure 4: Using MIL driver version 3.1x

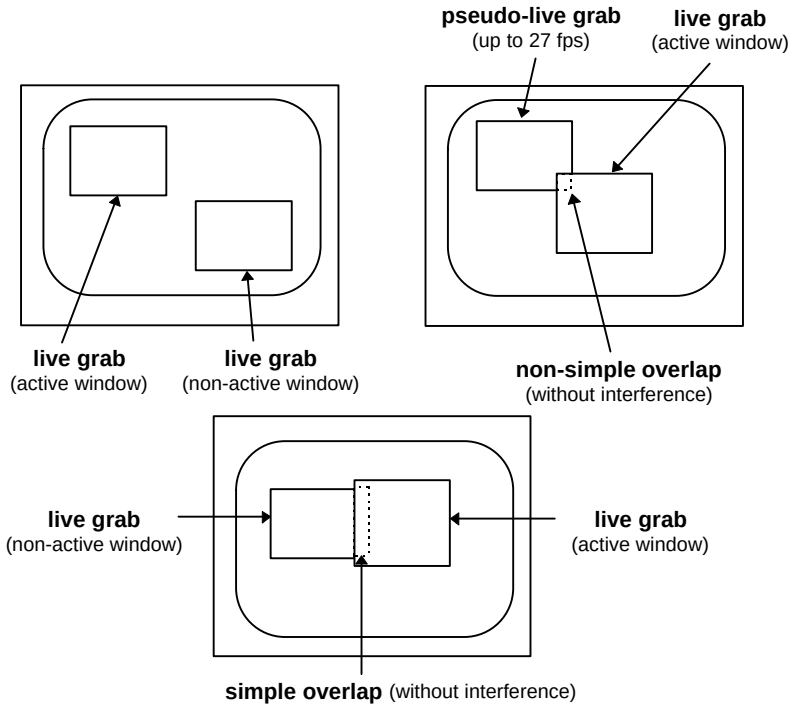

Figure 5: Using MIL driver version 4.0

SOFTWARE SUPPORT

Software support for the multiple boards is available through recent versions of the libraries, i.e. MIL-16/MIL-16 Lite 2.2 and MIL-32/MIL-32 Lite 3.1 or higher; and the recent versions of the drivers, i.e. MIL-16 Driver for Meteor 2.21 and MIL-32 driver for Meteor 3.11 or higher.

SOFTWARE EXAMPLE

Since there are now two boards being used in this example, there are two application identifiers. For each additional board introduced into the system, the addition of an identifier and allocation will be necessary. In this example, allocation of the first MIL system, digitizer, display, 2D data buffer is followed by the allocation of the second.

The function **MdispSelect()** outputs the specified image buffer contents to the specified display controller, while the function **MdigGrabContinuous** grabs continuously from the input device (the camera) until **MdigHalt(...)** is issued. Again there are two function calls for the two boards installed in the system. The function **MdispDeselect** stops displaying the specified image buffer, leaving the display blank. Deallocation of the buffer, display controller, digitizer, and system (**MbufFree, MdigFree, MdispFree, MsysFree** respectively) is done for each of the boards individually.

CODE EXAMPLE

The program following is an example that (using the defaults) displays live and pseudo-live images from multiple Matrox Meteor boards.

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

void main()
{
   MIL_ID MilApplication,   /* Application identifier. */

          MilSystem0,       /* System identifier.      */
          MilImage0,        /* Image identifier.       */
          MilDisplay0,      /* Display identifier.     */
          MilDigitizer0,    /* Digitizer identifier.   */

          MilSystem1,       /* System identifier.      */
          MilImage1,        /* Image identifier.       */
          MilDisplay1,      /* Display identifier.     */
          MilDigitizer1;    /* Digitizer identifier.   */
long   BufSizeX, BufSizeY;

   /* Allocation appplication .  */
   MappAlloc(M_DEFAULT, &MilApplication);

   /* Allocate first system objects.  */
   MsysAlloc(M_SYSTEM_METEOR, M_DEV0, M_DEFAULT, &MilSystem0);
   MdispAlloc(MilSystem0, M_DEV0, M_DEF_DISPLAY_FORMAT, M_DEFAULT, &MilDisplay0);
   MdigAlloc(MilSystem0, M_DEV0, M_DEF_DIGITIZER_FORMAT, M_DEFAULT, &MilDigitizer0);
   BufSizeX=MdigInquire(MilDigitizer0, M_SIZE_X, M_NULL);
   BufSizeY=MdigInquire(MilDigitizer0, M_SIZE_Y, M_NULL);
   MbufAlloc2d(MilSystem0, BufSizeX, BufSizeY, 8, M_IMAGE+M_DISP+M_PROC+M_GRAB, &MilImage0);
   MdispSelect(MilDisplay0, MilImage0);

   /* Allocate second system objects.  */
   MsysAlloc(M_SYSTEM_METEOR, M_DEV1, M_DEFAULT, &MilSystem1);
   MdispAlloc(MilSystem1, M_DEV0, M_DEF_DISPLAY_FORMAT, M_DEFAULT, &MilDisplay1);
   MdigAlloc(MilSystem1, M_DEV0, M_DEF_DIGITIZER_FORMAT, M_DEFAULT, &MilDigitizer1);
   BufSizeX=MdigInquire(MilDigitizer1, M_SIZE_X, M_NULL);
   BufSizeY=MdigInquire(MilDigitizer1, M_SIZE_Y, M_NULL);
   MbufAlloc2d(MilSystem1, BufSizeX, BufSizeY, 8, M_IMAGE+M_DISP+M_PROC+M_GRAB, &MilImage1);
   MdispSelect(MilDisplay1, MilImage1);

/* Grab continuously in both systems. */
   MdigGrabContinuous(MilDigitizer0, MilImage0);
   MdigGrabContinuous(MilDigitizer1, MilImage1);

/* When a key is pressed, halt. */
   printf("Continuous grab in progress. Adjust your cameras and\n");
   printf("press <Enter> to stop grabbing.\n");
   getchar();

/* Stop continuous grab. */
   MdigHalt(MilDigitizer1);
   MdigHalt(MilDigitizer0);
```

(Code continued)

```
    /* Pause to show the result. */
    printf("\nDisplaying the last grabbed images.\n");
    printf("Press <Enter> to end.\n");
    getchar();

    /* Free second system objects.  */
    MdispDeselect(MilDisplay1, MilImage1);
    MbufFree(MilImage1);
    MdigFree(MilDigitizer1);
    MdispFree(MilDisplay1);
    MsysFree(MilSystem1);

    /* Free first system objects.  */
    MdispDeselect(MilDisplay0, MilImage0);
    MbufFree(MilImage0);
    MdigFree(MilDigitizer0);
    MdispFree(MilDisplay0);
    MsysFree(MilSystem0);

    /* Free appplication .  */
    MappFree(MilApplication);
}
```

Corporate Headquarters:
Canada and U.S.A.
Matrox Electronic Systems Ltd.
1055 St.Regis Blvd.
Dorval, Quebec, Canada
H9P 2T4
Tel: (514) 685-7230
Fax: (514) 822-6273

Sales Offices:

U.K.
Matrox (UK) Ltd.
Sefton Park, Stoke Poges
Buckinghamshire
U.K. SL2 4JS
Tel: +44 (0) 1753 665500
Fax: +44 (0) 1753 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 GmbH
Inselkammerstr.8
D-82008 Unterhaching
Germany
Tel: 089/614 4740
Fax: 089/614 9743

Asia Pacific
Matrox Asia Liaison Office
Rm. 1901, 19/F, Workington Tower,
78 Bonham Strand E.,
Sheung Wan, Hong Kong.
Tel: 852.2877.5387
Fax: 852.2537.9530