MATMULT

Two dimensional graphical transformations, such as rotations, scalings, and translations are conveniently represented as homogeneous 3-by-3 matrices, which operate on homogeneous 3-vectors. Similarly, three dimensional graphical transformations are conveniently represented as homogeneous 4-by-4 matrices, which operate on homogeneous 4-vectors. MatMult provides utilities for creating and manipulating such matrices and vectors, and takes advantage of microcode support for high-speed 3-by-3 and 4-by-4 matrix multiplication.

All matrices and vectors in MatMult are represented as Common Lisp arrays of element type single-float, so the Common Lisp array functions are sufficient to create and access individual elements of these specialized arrays. However, MatMult provides convenient wrapper functions for most common operations on these arrays.

All the following functions that return arrays accept optional array arguments. If given a result argument, these functions alter the contents of that argument rather then allocating new storage. It is an error for the optional array argument to be not of element type single-float, or to have incorrect dimensions.

Requirements

MatMult should be run on an 1109 with a Weitek floating point chip set, but is also quite efficient on an 1186.

Installation

Load `MATMULT.LCOM` from the library.

Matrix Creation Functions

(MAKE-HOMOGENEOUS-3-VECTOR *X Y*)   [Function]

Returns a 3-vector of element type single-float. If *X* or *Y* is provided, then the corresponding element of the vector is set appropriately, otherwise it defaults to 0.0. The third element of the vector is always initialized to 1.0.

Note: Throughout this text, "set" is used to emphasize that the value of the result element is altered and that no new storage is allocated to it.

(MAKE-HOMOGENEOUS-3-BY-3 *&KEY A00 A01 A10 A20 A21*)   [Function]

Returns a 3-by-3 matrix of element type single-float. If a keyword argument is provided, the corresponding element of the matrix is set appropriately, otherwise entries default to 0.0. The (2 ,2) is always initialized to 1.0.

(MAKE-HOMOGENEOUS-N-BY-3 *N &KEY INITIAL-ELEMENT*)   [Function]

Returns an N-by-3 matrix of element type single-float. If the keyword argument is provided, all the elements in the first two columns are set appropriately, otherwise they default to 0.0. The third column is always initialized to 1.0.

(MAKE-HOMOGENEOUS-4-VECTOR   *X Y Z*)   [Function]

> Returns a 4-vector of element type single-float. If *X*, *Y* or *Z* is provided then the corresponding element of the vector is set appropriately, otherwise it defaults to 0.0. The forth element of the vector is always initialized to 1.0.

(MAKE-HOMOGENEOUS-4-BY-4   *&KEY A00 A01 A02 A03 A10 A11 A12   A13 A20 A21 A22 A23 A30 A31 A32*)   [Function]

> Returns a 4-by-4 matrix of element type single-float. If a keyword arguments is provided, the corresponding element of the matrix is set appropriately, otherwise entries default to 0.0. The (3 ,3) is always initialized to 1.0.

(MAKE-HOMOGENEOUS-N-BY-4 *N &KEY INITIAL-ELEMENT*)   [Function]

> Returns an N-by-4 matrix of element type single-float. If the keyword argument is provided, all the elements in the first three columns are set appropriately, otherwise they default to 0.0. The forth column is always initialized to 1.0.

(IDENTITY-3-BY-3   *RESULT*)   [Function]

> Returns a 3-by-3 identity matrix.

> If *RESULT* is supplied, it is side effected and returned.

> (That is, the storage associated with the optional result argument is reused for the result, rather than allocating new storage for the result.)

(IDENTITY-4-BY-4   *RESULT*)   [Function]

> Returns a 4-by-4 identity matrix. If *RESULT* is supplied, it is side effected and returned.

(ROTATE-3-BY-3 *RADIANS   RESULT*)   [Function]

> Returns a 3-by-3 rotation matrix specified by a counter-clockwise rotation of *RADIANS* radians. If *RESULT* is supplied, it is set and returned.

(ROTATE-4-BY-4-ABOUT-X *RADIANS   RESULT*)   [Function]

> Returns a 4-by-4 rotation matrix specified by a positive right-handed rotation of *RADIANS* radians about the X axis. If *RESULT* is supplied, it is set and returned.

(ROTATE-4-BY-4-ABOUT-Y *RADIANS   RESULT*)   [Function]

> Returns a 4-by-4 rotation matrix specified by a positive right-handed rotation of *RADIANS* radians about the Y axis. If *RESULT* is supplied, it is set and returned.

(ROTATE-4-BY-4-ABOUT-Z *RADIANS   RESULT*)   [Function]

> Returns a 4-by-4 rotation matrix specified by a positive right-handed rotation of *RADIANS* radians about the Z axis. If *RESULT* is supplied, it is set and returned.

(SCALE-3-BY-3 *SX SY   RESULT*)   [Function]

> Returns a 3-by-3 homogeneous scaling transformation that scales by a factor of *SX* along the X-axis and *SY* along the Y-axis. If *RESULT* is supplied, it is set and returned.

(SCALE-4-BY-4 *SX SY SZ RESULT*)   [Function]

>Returns a 4-by-4 homogeneous scaling transformation that scales by a factor of *SX* along the X-axis, *SY* along the Y-axis, and *SZ* along the Z axis. If *RESULT* is supplied, it is set and returned.

(TRANSLATE-3-BY-3 *TX TY RESULT*)   [Function]

>Returns a 3-by-3 homogeneous translation that translates by *TX* along the X-axis and *TY* along the Y-axis. If *RESULT* is supplied, it is set and returned.

(TRANSLATE-4-BY-4 *TX TY TZ RESULT*)   [Function]

>Returns a 4-by-4 homogeneous translation that translates by *TX* along the X-axis, *TY* along the Y-axis and *TZ* along the Z axis. If *RESULT* is supplied, it is set and returned.

(PERSPECTIVE-4-BY-4 *PX PY PZ RESULT*)   [Function]

>Returns a 4-by-4 homogeneous perspective transformation defined by *PX, PY*, and *PZ*. If *RESULT* is supplied, it is set and returned.

Matrix Multiplication Functions

If run on workstations equipped with the extended processor option, these functions make good use of the hardware floating-point unit. The three digits at the end of each function's name describe the dimensions of their arguments.

Note: The results of the following matrix multiplication functions are not guaranteed to be correct unless the matrix arguments are all different (Not EQ).

(MATMULT-133 *VECTOR MATRIX RESULT*)   [Function]

>Returns the inner product of a 3-vector, *VECTOR*, and a 3-by-3 matrix, *MATRIX*. If *RESULT* is supplied, it is set and returned.

(MATMULT-331 *MATRIX VECTOR RESULT*)   [Function]

>Returns the inner product of a 3-by-3 matrix, *MATRIX*, and a 3-vector, *VECTOR*. If *RESULT* is supplied, it is set and returned.

(MATMULT-333 *MATRIX-1 MATRIX-2 RESULT*)   [Function]

>Returns the inner product of a 3-by-3 matrix, *MATRIX-1*, and another 3-by-3 matrix, *MATRIX-2*. If *RESULT* is supplied, it is set and returned.

(MATMULT-N33 *MATRIX-1 MATRIX-2 RESULT*)   [Function]

>Returns the inner product of an N-by-3 matrix, *MATRIX-1*, and a 3-by-3 matrix, *MATRIX-2*. If *RESULT* is supplied, it is set and returned.

(MATMULT-144 *VECTOR MATRIX RESULT*)   [Function]

>Returns the inner product of a 4-vector, *VECTOR*, and a 4-by-4 matrix, *MATRIX*. If *RESULT* is supplied, it is set and returned.

(MATMULT-441 *MATRIX VECTOR RESULT*)   [Function]

>Returns the inner product of a 4-by-4 matrix, *MATRIX*, and a 4-vector, *VECTOR*. If *RESULT* is supplied, it is set and returned.

(MATMULT-444 *MATRIX-1 MATRIX-2 RESULT*)   [Function]

Returns the inner product of a 4-by-4 matrix, *MATRIX-1*, and another 4-by-4 matrix, *MATRIX-2*. If *RESULT* is supplied, it is set and returned.

(MATMULT-N44 *MATRIX-1 MATRIX-2 RESULT*)   [Function]

Returns the inner product of an N-by-4 matrix, *MATRIX-1*, and a 4-by-4 matrix, MATRIX-2. If *RESULT* is supplied, it is set and returned.

Miscellaneous Functions

(PROJECT-AND-FIX-3-VECTOR *3-VECTOR   2-VECTOR*)   [Function]

The homogeneous *3-VECTOR* is projected onto the X-Y plane, coerced to integer coordinates (rounding by truncation) and returned. If *2-VECTOR* is supplied, it is set and returned.

(PROJECT-AND-FIX-N-BY-3 *N-3-MATRIX N-2-MATRIX*)   [Function]

The homogeneous N-by-3 matrix, *N-3-MATRIX*, is projected onto the X-Y plane row-by-row, coerced to integer coordinates (rounding by truncation) and returned. If *N-2-MATRIX* is supplied, it is set and returned.

(PROJECT-AND-FIX-4-VECTOR *4-VECTOR 2-VECTOR*)   [Function]

The homogeneous 4-vector, *4-VECTOR*, is projected onto the X-Y plane, coerced to integer coordinates (rounding by truncation) and returned. If *2-VECTOR* is supplied, it is set and returned.

(PROJECT-AND-FIX-N-BY-4 *N-4-MATRIX N-2-MATRIX*)   [Function]

The homogeneous N-by-4 MATRIX, *N-3-MATRIX*, is projected onto the X-Y plane row-by-row, coerced to integer coordinates (rounding by truncation) and returned. If *N-2-MATRIX* is supplied, it is set and returned.

(DEGREES-TO-RADIANS *DEGREES*)   [Function]

Returns *DEGREES* converted to radians.

Limitations

MatMult is not intended as a general matrix manipulation package; it is specialized for the 3-by-3 and 4-by-4 cases.

Use CmlFloatArray for more general floating point array facilities.

Example

```
(* ; "Try (spiral)")


(CL:DEFUN SPIRAL (&OPTIONAL (WINDOW (CREATEW))
                          &AUX
                          (WIDTH (WINDOWPROP WINDOW ’WIDTH))
                          (HALF-WIDTH (QUOTIENT WIDTH 2))
                          (HEIGHT (WINDOWPROP WINDOW ’HEIGHT))
                          (HALF-HEIGHT (QUOTIENT HEIGHT 2))
                          (SCALE-FACTOR (CL:EXP (QUOTIENT
                                  (CL:LOG (QUOTIENT (MIN WIDTH HEIGHT) 2.0)) 1440.0))))
   (LET ((LINE-1 (MAKE-HOMOGENEOUS-3-VECTOR 1.0 0.0))
         (LINE-2 (MAKE-HOMOGENEOUS-3-VECTOR))
         (TEMP (MAKE-HOMOGENEOUS-3-VECTOR))
         (POINTS (CL:MAKE-ARRAY 2))
         (TRANSFORM (MATMULT-333 (ROTATE-3-BY-3 (DEGREES-TO-RADIANS 2.5))
                         (SCALE-3-BY-3 SCALE-FACTOR SCALE-FACTOR)))
         (TRANSLATION (TRANSLATE-3-BY-3 HALF-WIDTH HALF-HEIGHT)))
        (CL:DO ((L-1 LINE-1)
                (L-2 LINE-2)
                (I 0 (CL:1+ I)))
               ((EQ I 1728))
               (MATMULT-133 L-1 TRANSFORM L-2)
               (MATMULT-133 L-2 TRANSLATION TEMP)
               (PROJECT-AND-FIX-3-VECTOR TEMP POINTS)
               (DRAWLINE HALF-WIDTH HALF-HEIGHT (CL:AREF POINTS 0)
                     (CL:AREF POINTS 1)
                     1
                     ’REPLACE WINDOW)
               (CL:ROTATEF L-1 L-2))))
```

[This page intentionally left blank]