Matrices

Computer Graphics - SET08116

EDINBURGH NAPIER UNIVERSITY

Outline

1. Matrices
2. Matrix Multiplication
3. Matrix Inverses

What is a Matrix?

A matrix is a $m \times n$ array of numbers, having n rows and m columns. If $n = m$, then the matrix **M** is said to be square.

$$\mathbf{M} = \begin{bmatrix} M_{11} & M_{12} & M_{13} & M_{14} \\ M_{21} & M_{22} & M_{23} & M_{24} \\ M_{31} & M_{32} & M_{33} & M_{34} \\ M_{41} & M_{42} & M_{43} & M_{44} \end{bmatrix}$$

We can denote the individual elements of the array using a subscript notation M_{ij} where i refers to the row of the matrix, and j refers to the column.

What is a Matrix?

The entries of the matrix where $i = j$ are called the *main diagonal* entries. If a square matrix only has non-zero values in its main diagonals, it is called a *diagonal matrix*.

The *transpose* of a matrix ($\mathbf{M}^T$) that is $m \times n$ becomes an $n \times m$ matrix where $\mathbf{M}_{ij}{}^T = \mathbf{M}_{ji}$.

We can think of vectors as just $n \times 1$ vectors. This is useful, as we will be transforming vectors using this approach.

Scaling a Matrix

As you can imagine, we can scale a matrix using standard techni

$$a\mathbf{M} = \mathbf{M}a = \begin{bmatrix} aM_{11} & aM_{12} & \cdots & aM_{1m} \\ aM_{21} & aM_{22} & \cdots & aM_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ aM_{n1} & aM_{n2} & \cdots & aM_{nm} \end{bmatrix}$$

Adding matrices also follows a simple pattern:

$$\mathbf{F} + \mathbf{G} = \begin{bmatrix} F_{11} + G_{11} & F_{12} + G_{12} & \cdots & F_{1m} + G_{1m} \\ F_{21} + G_{21} & F_{22} + G_{22} & \cdots & F_{2m} + G_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ F_{n1} + G_{n1} & F_{n2} + G_{n2} & \cdots & F_{nm} + G_{nm} \end{bmatrix}$$

Matrix Multiplication

We can also multiply two matrices together. This is a very importa... concept, and we use it to underpin most of the heavy work done on the GPU (for example - we multiply vectors by the model-view-projection matrix).

To multiply two matrices, the number of columns in the first matrix must equal the number of rows in the second. That is, we can multiply a $n \times m$ matrix by a $m \times p$ matrix. This produces an $n \times p$ matrix.

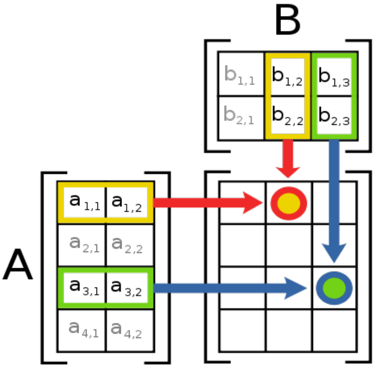
To multiply matrices **F** and **G** together, we use the following:

$$(\mathbf{FG})_{ij} = \sum_{k=1}^{m} F_{ik} G_{kj}$$

Matrix Multiplication

There is a simpler method of looking at this. We are essentially ta

dot product of i-th row of **F** and the j-th column of **G**. The following diagram from Wikipedia should help:



Identity Matrix

When we have been building matrices, we have been using the command mat4(1.0f) to begin with. This command allows us to build what is called an *identity matrix*. An identity matrix is a $n \times n$ diagonal matrix with 1 on the main diagonals:

$$\mathbf{I}_n = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}$$

When a matrix is multiplied by an identity matrix, the original matrix is the result. That is:

$$\mathbf{M}\mathbf{I}_n = \mathbf{I}_n\mathbf{M} = \mathbf{M}$$

Matrix Inverses

We say that an $n \times n$ matrix is invertible if and only if there exists a matrix M^{-1} such that:

$$\mathbf{M}\mathbf{M}^{-1} = \mathbf{M}^{-1}\mathbf{M} = \mathbf{I}$$

We say that $\mathbf{M}^{-1}$ is the *inverse* of $\mathbf{M}$. Remember, we have been using the inverse operation on a matrix during our lighting calculations.

Any matrix that does not have an inverse is called *singular*. An example of a singular matrix is any one that has all zeros for any row or column. The matrices we typically build for transformation operations never have this property.

Matrix Inverses

Two important properties to note about matrix inverse.

A matrix $\mathbf{M}$ is invertible if and only if $\mathbf{M}^{-1}$ is invertible.

If $\mathbf{F}$ and $\mathbf{G}$ are $n \times n$ invertible matrices, then their product $\mathbf{FG}$ is also invertible. The result is defined as follows:

$$(\mathbf{FG})^{-1} = \mathbf{G}^{-1}\mathbf{F}^{-1}$$

Constructing an Inverse Matrix

We would use the relevant libraries provided to actually construct an inverse matrix, but it is worth knowing how it can be done.

We will use an algorithm know as *Gauss-Jordon* elimination to perform this.

First, given a $n \times n$ matrix, $\mathbf{M}$, we must generate a new $n \times 2n$ matrix, $\tilde{\mathbf{M}}$, which looks as follows:

$$\tilde{\mathbf{M}} = \begin{bmatrix} M_{11} & M_{12} & \cdots & M_{1n} & 1 & 0 & \cdots & 0 \\ M_{21} & M_{22} & \cdots & M_{2n} & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ M_{m1} & M_{m2} & \cdots & M_{mn} & 0 & 0 & \cdots & 1 \end{bmatrix}$$

Constructing an Inverse Matrix

The rest of the algorithm goes as follows ($\tilde{\mathbf{M}}$ always refers to the current state, not the original):

1. Set the column j equal to 1. We will loop through columns 1 to n.
2. Find the row i with $i \geq j$ such that $\tilde{M}_{ij}$ has the largest absolute value. If no such row exists for which $\tilde{M}_{ij} \neq 0$ then $\mathbf{M}$ is not invertible.
3. If $i \neq j$, then exchange rows i and j. This operation will remove zeros from the main diagonal.
4. Multiply row j by $1/\tilde{M}_{jj}$. This sets $\tilde{M}_{jj} = 1$.
5. For each row r where $1 \leq r \leq n$ and $r \neq j$, add $-\tilde{M}_{rj}$ times row j to row r.
6. If $j < n$, increment j and loop to step 2.

Recommended Reading

Chapter 3 of the maths text goes into more detail on matrices. In particular, matrix determinants and linear systems are discussed.