



Raben Systems, Inc.

[Home](#)

Linear algebra with OpenCV

Tue, 11/20/2012 - 10:22 — Vern

The [OpenCV image processing libraries](#) provides hundreds of computer vision algorithms useful in a wide variety of imaging applications. The "core" of the library supports numerous matrix operations which are useful for processing images and for solving linear equations.

A matrix is just a rectangular array of numbers arranged in rows and columns. Here is an example of how to load values into a matrix with 2 rows and 2 columns and add them:

```
#include <iostream>
#include <opencv2/core/core.hpp>
using namespace std;
using namespace cv;

void addMatrixExample() {
    float a[2][2] = {{1, 2},
                     {3, 4}};
    float b[2][2] = {{5, 6},
                     {7, 8}};

    Mat A = Mat(2, 2, CV_32FC1, a);
    Mat B = Mat(2, 2, CV_32FC1, b);
    Mat C;
    C = A + B;
    cout << "C =" << endl << " " << C << endl << endl;
}

int main() {
```

```
addMatrixExample();  
}
```

This will output:

C =
[6, 8;
10, 12]
to the console.

Multiplication of matrices is easy as well. Remember that the rows in the first matrix are multiplied by the columns in second matrix. You'll get an assert error from OpenCV if the number of columns in matrix A doesn't match the number of rows in matrix B.

```
void matrixMultiplication(){  
    float a[2][3] = {{ 3, 1, 2},  
                     {-2, 0, 5}};  
    float b[3][2] = {{-1, 3},  
                     {0, 5},  
                     {2, 5}};  
    Mat A = Mat(2, 3, CV_32FC1, a);  
    Mat B = Mat(3, 2, CV_32FC1, b);  
    Mat C;  
    C = A * B;  
    cout << "C = A * B" << endl << " " << C << endl << endl;  
    C = B * A;  
    cout << "C = B * A" << endl << " " << C << endl << endl;  
}
```

This will output:

C = A * B :
[1, 24;
12, 19]

C=B*A:
[-9, -1, 13;

-10, 0, 25;
-4, 2, 29]

To solve a linear equation such as:

$$3x + 2y = 7$$

$$-6x + 6y = 6$$

We start by representing the equation in matrix form:

$$AX = B$$

The elements of matrix A are set to the coefficients of the variables on the left hand side of the equation and matrix B is set to the right hand values.

To solve we simply take the inverse of matrix A times matrix B

$$X = A^{-1}B$$

```
void solveLinearEquation() {  
    // 3x + 2y = 7  
    // -6x + 6y = 6  
    float a[2][2] = {{ 3, 2},  
                     {-6, 6}};  
  
    float b[2][1] = { {7},  
                     {6} };  
  
    Mat A = Mat(2,2 ,CV_32FC1, a);  
    Mat B = Mat(2,1, CV_32FC1, b);  
  
    Mat x = A.inv() * B;  
    cout << "x=" << endl << " " << x << endl;  
}
```

This will output the solution:

$x =$

$[1; 2]$

to the console.

If your memories of linear algebra class have faded (like mine have), you might take it look at the some of the excellent videos that the KhanAcademy has online at

<http://www.khanacademy.org/math/linear-algebra> .

Powered by [Drupal](#)