Gaussian Mask

Comp Intel
Holger Findling

Software Implementation

The following C++ code calculates the Gaussian mask for 0.5, 1.0, 1.5, and 2.0 sigma. Expanding the matrix for larger sigma values is left as an exercise.

Gaussian.cpp

```
#include "stdafx.h"
#include "GaussianMatrix.h"
#include <stdio.h>

using namespace System;

void PrintTopHalf(GaussianMatrix G)
{
	for (int j = 0; j <= 6; j++) {
		printf("%3d %3d %3d %3d %3d %3d %3d %3d %3d %3d %3d %3d %3d \n",
			G.ConMask[j][0],
			G.ConMask[j][1],
			G.ConMask[j][2],
			G.ConMask[j][3],
			G.ConMask[j][4],
			G.ConMask[j][5],
			G.ConMask[j][6],
			G.ConMask[j][7],
			G.ConMask[j][8],
			G.ConMask[j][9],
			G.ConMask[j][10],
			G.ConMask[j][11],
			G.ConMask[j][12]);
	}
	return;
}


void PrintBottomHalf(GaussianMatrix G)
{
	for (int j = 5; j >= 0; j--) {
		printf("%3d %3d %3d %3d %3d %3d %3d %3d %3d %3d %3d %3d %3d \n",
			G.ConMask[j][0],
			G.ConMask[j][1],
			G.ConMask[j][2],
			G.ConMask[j][3],
			G.ConMask[j][4],
			G.ConMask[j][5],
			G.ConMask[j][6],
			G.ConMask[j][7],
			G.ConMask[j][8],
```

```
                    G.ConMask[j][9],
                    G.ConMask[j][10],
                    G.ConMask[j][11],
                    G.ConMask[j][12]);
        }
        printf("\n\n");
        return;
}


int main(array<System::String ^> ^args)
{
   GaussianMatrix G;

   // Gaussian Mask 0.5 Sigma
   G.CalculateMask(0.5);
   PrintTopHalf(G);
   PrintBottomHalf(G);


   // Gaussian Mask 1.0 Sigma
   G.CalculateMask(1.0);
   PrintTopHalf(G);
   PrintBottomHalf(G);

   // Gaussian Mask 1.5 Sigma
   G.CalculateMask(1.5);
   PrintTopHalf(G);
   PrintBottomHalf(G);

   // Gaussian Mask 2.0 Sigma
   G.CalculateMask(2.0);
   PrintTopHalf(G);
   PrintBottomHalf(G);


   Console::Read();
   return 0;
}
```

GaussianMatrix.h

```
#pragma once
class GaussianMatrix
{
public:
   GaussianMatrix(void);
   ~GaussianMatrix(void);

   void CalculateMask(double sigma);

   double Mask[13][13];
   int ConMask[13][13];

};
```

GaussianMatrix.cpp

```
#include "StdAfx.h"
#include "GaussianMatrix.h"
#include <cmath>

using namespace System;

GaussianMatrix::GaussianMatrix(void)
{
}

GaussianMatrix::~GaussianMatrix(void)
{
}

void GaussianMatrix::CalculateMask(double sigma)
{
   int x = 0;
   int y = 0;
   double term1 = 1.0 / (2.0 * 3.14 * sigma * sigma);

   for (int j = 6; j >= 0; j--, y++ )
   {
      for (int i = -6; i <= 6; i++, x++ )
      {
         double sqr = (i * i) + (j * j);
         double term2 = (sqr/ (2 * sigma * sigma)) * -1.0;
         Mask[y][x] = term1 * exp(term2);
      }
      x = 0;
   }

   double ratio = 100.0 / Mask[6][6];
   for (int j = 0; j <= 6; j++) {
      for (int i = 0; i <= 12; i++) {
         ConMask[j][i] = (int) Math::Round (Mask[j][i] * ratio);
   }}
}
```

Gaussian Masks

0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	2	14	2	0	0	0	0	0
0	0	0	0	0	14	100	14	0	0	0	0	0
0	0	0	0	0	2	14	2	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0

Gaussian Mask – 0.5 Sigma

0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	1	1	0	0	0	0	0
0	0	0	0	2	8	14	8	2	0	0	0	0
0	0	0	1	8	37	61	37	8	1	0	0	0
0	0	0	1	14	61	100	61	14	1	0	0	0
0	0	0	1	8	37	61	37	8	1	0	0	0
0	0	0	0	2	8	14	8	2	0	0	0	0
0	0	0	0	0	1	1	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0

Gaussian Mask – 1.0 Sigma

0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	2	3	2	1	0	0	0	0
0	0	0	2	6	11	14	11	6	2	0	0	0
0	0	1	6	17	33	41	33	17	6	1	0	0
0	0	2	11	33	64	80	64	33	11	2	0	0
0	0	3	14	41	80	100	80	41	14	3	0	0
0	0	2	11	33	64	80	64	33	11	2	0	0
0	0	1	6	17	33	41	33	17	6	1	0	0
0	0	0	2	6	11	14	11	6	2	0	0	0
0	0	0	0	1	2	3	2	1	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0

Gaussian Mask – 1.5 Sigma

0	0	0	0	1	1	1	1	1	0	0	0	0
0	0	1	1	3	4	4	4	3	1	1	0	0
0	1	2	4	8	12	14	12	8	4	2	1	0
0	1	4	11	20	29	32	29	20	11	4	1	0
1	3	8	20	37	54	61	54	37	20	8	3	1
1	4	12	29	54	78	88	78	54	29	12	4	1
1	4	14	32	61	88	100	88	61	32	14	4	1
1	4	12	29	54	78	88	78	54	29	12	4	1
1	3	8	20	37	54	61	54	37	20	8	3	1
0	1	4	11	20	29	32	29	20	11	4	1	0
0	1	2	4	8	12	14	12	8	4	2	1	0
0	0	1	1	3	4	4	4	3	1	1	0	0
0	0	0	0	1	1	1	1	1	0	0	0	0

Gaussian Mask – 2.0 Sigma