

# Gaussian Kernel Smoothing

Moo K. Chung  
mchung@stat.wisc.edu

July 30, 2012

## 1 Kernel Smoothing

Gaussian kernel smoothing most widely used image smoothing technique in brain imaging. Consider the integral transform

$$Y(t) = \int K(t, s)X(s) ds, \quad (1)$$

where  $K$  is the *kernel* of the integral. Given the input signal  $X$ ,  $Y$  represents the output signal. The smoothness of the output depends on the smoothness of the kernel. We assume the kernel to be unimodal and isotropic. When the kernel is isotropic, it has radial symmetry and should be invariant under rotation. So it has the form

$$K(t, s) = f(\|t - s\|)$$

for some smooth function  $f$ . Since the kernel only depends on the difference of the arguments, with the abuse of notation, we can simply write  $K$  as

$$K(t, s) = K(t - s).$$

We may further assume the kernel is normalized such that

$$\int K(t) dt = 1.$$

With this specific form of kernel  $K$ , (1) can be written as

$$Y(t) = K * X(t) = \int K(t - s)X(s) ds. \quad (2)$$

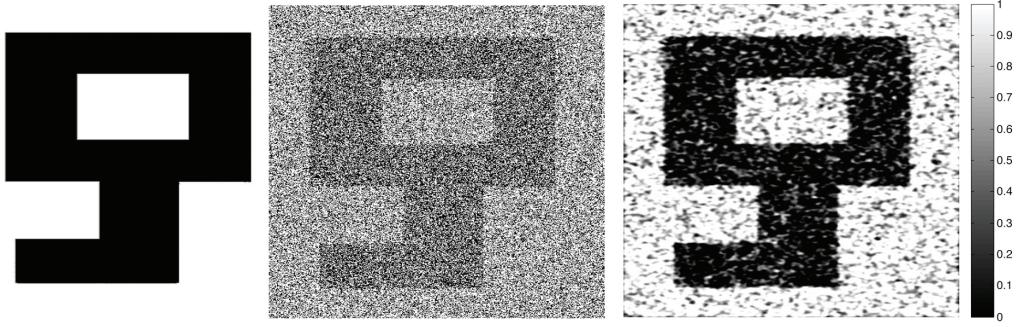


Figure 1: Illustration of Gaussian kernel smoothing as a form of signal detection. Gaussian white noise  $N(0, 2^2)$  is added to the binary image. Kernel smoothing with Gaussian kernel  $K_2$  is applied to the noise image to recover the original shape.

The integral transform (2) is called *kernel smoothing*. We may assume further that  $K$  is dependent on *bandwidth*  $\sigma$ . The bandwidth will determine the spread of kernel weights such that

$$\begin{aligned}\lim_{\sigma \rightarrow \infty} K(t, s; \sigma) &= 1 \\ \lim_{\sigma \rightarrow 0} K(t, s; \sigma) &= \delta(t - s),\end{aligned}$$

where  $\delta$  is the Dirac-delta function [1, 2]. The Dirac-delta function is a special case of *generalized functions* [3, 12]. The Dirac-delta function is usually traditionally defined as

$$\delta(t) = 0 \text{ if } t \neq 0, \quad \delta(t) dt = 1.$$

This is also referred to as the *impulse function* in literature. Figure 1 illustrates Gaussian kernel smoothing in a simple toy example.

## 2 Gaussian Kernel Smoothing

All brain images are inherently noisy due to errors associated with image acquisition. Compounding the image acquisition errors, there are errors caused by image registration and segmentation. So it is necessary to smooth out the segmented images before any statistical analysis is performed to boost statistical power. Among

many possible image smoothing methods [5, 8], *Gaussian kernel smoothing* has emerged as a de facto smoothing technique in brain imaging.

The *Gaussian kernel* in 1D is defined as

$$K(t) = \frac{1}{\sqrt{2\pi}} e^{t^2/2}.$$

Let's scale the Gaussian kernel  $K$  by the bandwidth  $\sigma$ :

$$K_\sigma(t) = \frac{1}{\sigma} K\left(\frac{t}{\sigma}\right).$$

This is the density function of the normal distribution with mean 0 and variance  $\sigma^2$ .

The  $n$ -dimensional isotropic Gaussian kernel is defined as the product of  $n$  1D kernels. Let  $t = (t_1, \dots, t_n)' \in \mathbb{R}^n$ . Then the  $n$ -dimensional kernel is given by

$$K_\sigma(t) = K_\sigma(t_1)K_\sigma(t_2)\cdots K_\sigma(t_n) \quad (3)$$

$$= \frac{1}{(2\pi)^{n/2}\sigma^n} \exp\left(\frac{1}{2\sigma^2} \sum_{i=1}^n t_i^2\right). \quad (4)$$

Subsequently,  $n$ -dimensional isotropic Gaussian kernel smoothing  $K_\sigma * X$  can be done by applying 1-dimensional smoothing  $n$  times in each direction by factorizing the kernel as

$$\begin{aligned} K_\sigma * X(t) &= \int K_\sigma(t-s)X(s) ds \\ &= \int K_\sigma(t_1-s_1)K_\sigma(t_2-s_2)\cdots K_\sigma(t_n-s_n)X(s) ds \\ &= \int K_\sigma(t_1-s_1)\cdots K_\sigma(t_{n-1}-s_{n-1}) ds_1\cdots ds_{n-1} \\ &\quad \times \int K_\sigma(t_n-s_n)X(s) ds_n. \end{aligned}$$

Note that  $K_\sigma * X$  is the scale-space representation of image  $X$  first introduced in [13]. Each  $K_\sigma * X$  for different values of  $\sigma$  produces a blurred copy of its original. The resulting scale-space representation from coarse to fine resolution can be used in multiscale approaches such as hierarchical searches and image segmentation. See [6], [9], [10], [11], and [14] for the review of the major problems in scale-space and multiscale descriptions of images.

### 3 FWHM

The *bandwidth*  $\sigma$  defines the spread of kernel. In brain imaging, the spread of kernel is usually measured in terms of the *full width at the half maximum* (FWHM) of Gaussian kernel  $K_\sigma$ , which is given by  $2\sqrt{2 \ln 2}\sigma$ . This can be easily seen by representing the kernel using the radius  $r^2 = x_1^2 + \dots + x_n^2$ :

$$K_\sigma(r) = \frac{1}{(2\pi)^{n/2}\sigma^n} \exp\left(-\frac{r^2}{2\sigma^2}\right).$$

The peak of the kernel is  $K_\sigma(0)$ . Then the *full width at half maximum* (FWHM) of the peak is given by

$$\text{FWHM} = 2\sqrt{2 \ln 2}\sigma.$$

The FWHM increases linearly as  $\sigma$  increases in Euclidean space.

### 4 MATLAB Implementation

Suppose that noisy observation is obtained at each grid point  $(\frac{i}{100}, \frac{j}{100}) \in [0, 1]^2$ . This basically forms 2D image. The signal  $\mu$  is assumed to be

$$\mu(t_1, t_2) = \cos(10t_1) + \sin(10t_2).$$

The signal is contaminated by Gaussian white noise  $\epsilon \sim N(0, 0.4^2)$ . Using `meshgrid` command, we generate  $101 \times 101$  2D grid points. The grid coordinates are stored in `px` and `py`, which are  $101 \times 101$  matrix. Figure 2 shows the simulated image  $Y = \mu + \epsilon$ .

```
[px,py] = meshgrid([0:0.01:1])  
  
px(1:3,1:3)  
ans =  
    0    0.0100    0.0200  
    0    0.0100    0.0200  
    0    0.0100    0.0200  
    ...  
  
py(1:3,1:3)  
ans =
```

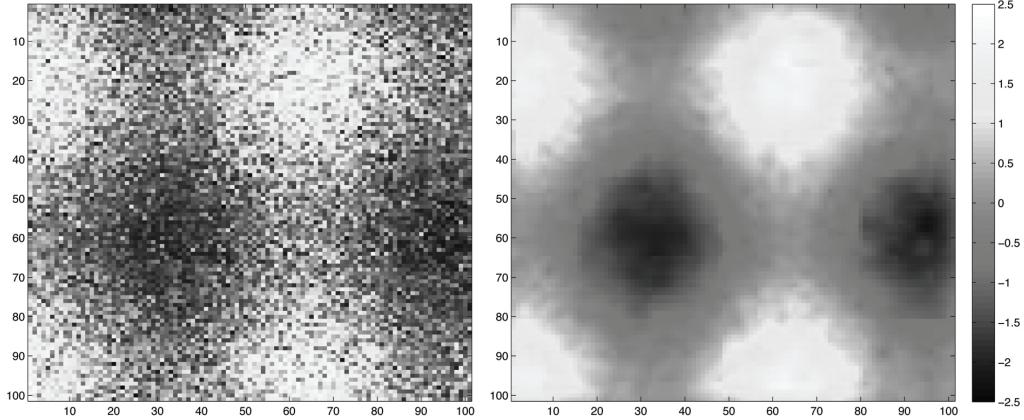


Figure 2: Left: simulated noise  $N(0, 0.4^2)$  is added to signal  $\mu(t_1, t_2) = \cos(10t_1) + \sin(10t_2)$ . Right: Gaussian kernel smoothing is applied with bandwidth 1.

```

0          0          0
0.0100    0.0100    0.0100
0.0200    0.0200    0.0200
...

```

```

mu=cos (10*px)+sin (8*py);
e=normrnd (0, 0.4, 101, 101);
Y=mu+e;
figure; imagesc(Y); colorbar;

```

The simulated image  $Y$  is fairly noisy. Gaussian kernel smoothing with bandwidth 1 is applied to  $Y$  to increase smoothness. The Gaussian kernel is constructed using `inline` function.

```
K=inline ('exp(-(x.^2+y.^2)/2/sig^2)')
```

```
K =
```

```

Inline function:
K(sig,x,y) = exp(-(x.^2+y.^2)/2/sig^2)

```

The inline function  $K$  has three arguments: two coordinates  $x$  and  $y$  and bandwidth  $\sigma$ . The kernel is then constructed discretely using  $5 \times 5$  grid and then

renormalizing it such that the kernel sums up to 1. For small `sig`, the discrete kernel weights are focused in the center of the  $5 \times 5$  window while for large `sig`, the kernel weights are more dispersed. Error will increase if a smaller window is used. Smoothing is done by the convolution `conv2`. The smoothed image is in Figure 2.

```
[dx,dy]=meshgrid([-2:2]);
weight=K(0.5,dx,dy)/sum(sum(K(0.5,dx,dy)) )

weight =
0.0000 0.0000 0.0002 0.0000 0.0000
0.0000 0.0113 0.0837 0.0113 0.0000
0.0002 0.0837 0.6187 0.0837 0.0002
0.0000 0.0113 0.0837 0.0113 0.0000
0.0000 0.0000 0.0002 0.0000 0.0000

weight=K(1,dx,dy)/sum(sum(K(1,dx,dy)) )

weight =
0.0030 0.0133 0.0219 0.0133 0.0030
0.0133 0.0596 0.0983 0.0596 0.0133
0.0219 0.0983 0.1621 0.0983 0.0219
0.0133 0.0596 0.0983 0.0596 0.0133
0.0030 0.0133 0.0219 0.0133 0.0030

Ysmooth=conv2(Y,weight,'same');
figure; imagesc(Ysmooth); colorbar;
```

## 5 Smoothing of DWI Stroke Lesions

The sample diffusion-weighted image (DWI) is obtained from Dong-Eog Kim of Dongguk University Hospital in Korea as a part of image-based stroke registry. See [4] for details on the image. In DWI, each voxel has an image intensity that reflects a single best measurement of the rate of water diffusion at that location. DTI can be used to identify severely ischemic brain regions after stroke onset than more traditional T1 or T2 MRI [7].

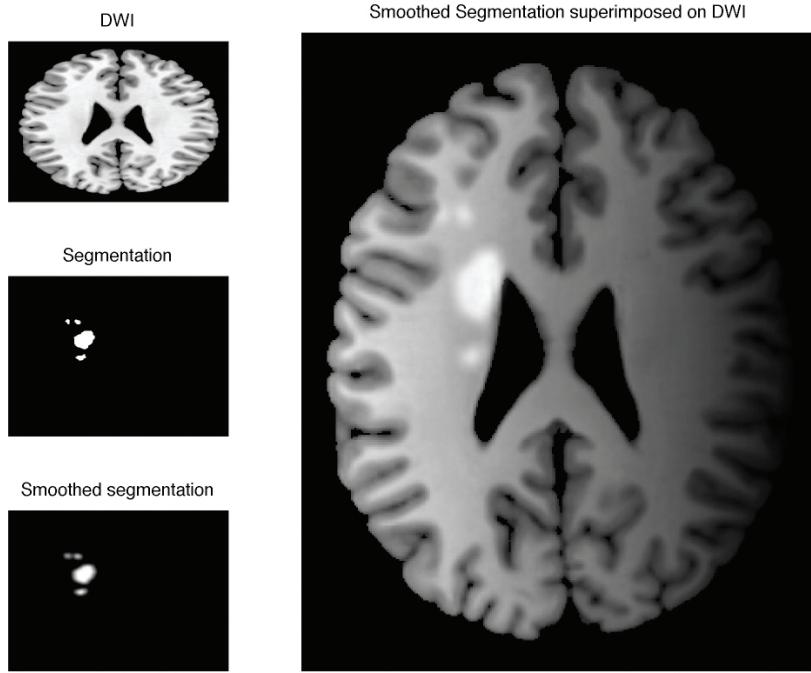


Figure 3: Diffusion-weighted image (DWI) and the segmentation of ischemic regions. The segmented region is smoothed with Gaussian kernel and superimposed on DWI. Smoothing is necessary before group analysis in reducing the regions of false positives.

One slice of sample DWI is stored as bitmap image format as `DWI.bmp`, which can be read using the built-in MATLAB function `imread`. Stroke lesions in DWI is segmented by a human expert and stored as a binary file `DWI-segmentation.bmp`.

```

DWimg= 'DWI';
DWI=imread(DWimg,'bmp');

segmentation = 'DWI-segmentation';
f=imread(segmentation,'bmp');

```

Once we read image files, we are interested in smoothing the binary segmentation using the Gaussian kernel smoothing procedure. Bitmap files consists of a matrix of three columns representing red, green and blue colors. So we simply

take out the first column and normalize it to have pixel values 0 or 1. The normalized segmentation is then smoothed using Gaussian kernel smoothing, which is implemented as `gaussblur2D.m`. The first argument to function `gaussblur2D` is the 2D array to smooth out and the second argument is the bandwidth given in terms of full width at the half maximum (FWHM) of the kernel.

```
binarize = double(f(:,:,1)/255);
fblur= gaussblur2D(binarize,10);

n=size(fblur);
fblurout = zeros(n(1), n(2), 3);
fblurout (:,:,1)=fblur;
fblurout (:,:,3)=fblur;
```

For visualization, we use subplot consisting of  $3 \times 3$  image grid. Four different images are displayed by partitioning the  $3 \times 3$  image grid. We have superimposed the smoothed segmentation on top of DWI by specifying the transparency of each pixel using `set` command with '`AlphaData`' option. Figure 3 is obtained by running the codes below.

```
figure; subplot(3,3,1);
imagesc(DWI);
axis off
title('DWI')

subplot(3,3,4);
imagesc(f);
axis off
title('Segmentation')

subplot(3,3,7); imagesc(fblurout)
axis off
title('Smoothed segmentation')

subplot(3,3,[2:3, 8:9]);
imagesc(DWI);
axis off
hold on
```

```

imgAlpha = repmat(0:1/n(2):1-1/n(2),n(1),1);
img = imagesc(fblurout);
set(img,'AlphaData',imgAlpha);
title('Smoothed Segmentation superimposed on DWI')

```

## References

- [1] N. Berline, E. Getzler, and Vergne M. *Heat kernels and dirac operators*. Springer-Verlag, 1991.
- [2] P.A.M. Dirac. *The principles of quantum mechanics*. Oxford University Press, USA, 1981.
- [3] I.M. Gel'fand, G.E. Shilov, and E. Saletan. *Generalized functions*. Academic Press New York, 1964.
- [4] D.-E. Kim, K.-J. Park, D. Schellingerhout, S.-W. Jeong, M.-G. Ji, W.J. Choi, Y.-O. Tak, G.-H. Kwan, E.A. Koh, S.-M. Noh, H.Y. Jang, T.-Y. Kim, J.-W. Jeong, J.S. Lee, and Choi H.-K. A new image-based stroke registry containing quantitative magnetic resonance imaging data. *Cerebrovascular Diseases*, 32:567–576, 2011.
- [5] S. Kovačić and R. Bajcsy. Multiscale/multiresolution representations. *Brain Warping*, pages 45–65, 1999.
- [6] T. Lindeberg. *Scale-Space Theory in Computer Vision*. Kluwer Academic Publisher, 1994.
- [7] T. Neumann-Haefelin, H.J. Wittsack, F. Wenerski, M. Siebler, R.J. Seitz, U. Mödder, and H.J. Freund. Diffusion-and perfusion-weighted mri: the dwi/pwi mismatch region in acute stroke. *Stroke*, 30:1591–1597, 1999.
- [8] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 12:629–639, 1990.
- [9] J-B Poline and B.M. Mazoyer. Analysis of individual brain activation maps using hierarchical description and multiscale detection. *IEEE Transactions on Medical Imaging*, 13:702–710, 1994.

- [10] J-B Poline, K.J. Worsley, Frackowiak R.S.J. Holmes, A.P., and K.J. Friston. Estimating smoothness in statistical parametric maps: Variability of  $p$  values. *Journal of Computer Assisted Tomography*, 19:788–796, 1995.
- [11] D.O. Siegmund and K.J. Worsley. Testing for a signal with unknown location and scale in a stationary gaussian random field. *Annals of Statistics*, 23:608–639, 1996.
- [12] I. Stakgold. *Boundary value problems of mathematical physics*. Society for Industrial Mathematics, 2000.
- [13] A. Witkin. Scale-space filtering. In *Int. Joint Conference on Artificial Intelligence*, pages 1019–1021, 1983.
- [14] K.J. Worsley, S. Marrett, P. Neelin, and A.C. Evans. Searching scale space for activation in pet images. *Human Brain Mapping*, 4:74–90, 1996.