

Filtrace obrazu ve frekvenční oboru

Václav Hlaváč

České vysoké učení technické v Praze

Český institut informatiky, robotiky a kybernetiky

160 00 Praha 6, Jugoslávských partyzánů 1580/3

<http://people.ciirc.cvut.cz/hlavac>, vaclav.hlavac@cvut.cz

také z Centra strojového vnímání, <http://cmp.felk.cvut.cz>

Osnova přednášky:

- ◆ Konvoluce jako filtrace ve frekvenční oblasti.
- ◆ Příklady filtrace dolní propustí: ostré ořezání; hladký Gaussián.
- ◆ Příklady filtrace horní propustí: ostré ořezání; hladký Gaussián.
- ◆ Butterworthův filtr.
- ◆ Homomorfická filtrace oddělující osvětlení a odrazivost.
- ◆ Systematický návrh 2D FIR filtrů.

Filtrace ve frekvenčním oboru

1. $F(u, v) = \mathcal{F}\{f(x, y)\}$

2. $G(u, v) = H(u, v) \cdot * F(u, v),$

kde $\cdot *$ znamená násobení sobě odpovídajících prvků v maticích.

3. $g(x, y) = \mathcal{F}^{-1}\{G(u, v)\}$

Připomínka pro cvičení: Pro zobrazování obvykle používáme $\ln \|F(u, v)\|$, abychom snížili velký dynamický rozsah. Při filtraci se ale musí pracovat s původním spektrem $F(u, v)$.

Konvoluce ve frekvenčním spektru

- ◆ Násobení v matici prvek po prvku.
- ◆ (Vysoká) rychlost je dána využitím rychlé FFT.

Mějme matici a rozměru $M \times N$ a matici b rozměru $P \times Q$.

Konvoluci $c = a * b$ lze spočítat postupem:

1. Dopln matice a , b nulami, aby měly rozměr alespoň $M + P - 1$, $N + Q - 1$ (obvykle až do velikosti mocniny dvou kvůli FFT).
2. Vypočítej 2D FFT matic a , b (v MATLABu pomocí `fft2`). Výsledkem jsou matice A , B .
3. Vynásob komplexní Fourierova spektra $C = A . * B$, součin prvek po prvku.
4. Výsledek konvoluce c získej inverzní Fourierovu transformací (v MATLABu pomocí `ifft2`).

Příklad 2D konvoluce v MATLABu

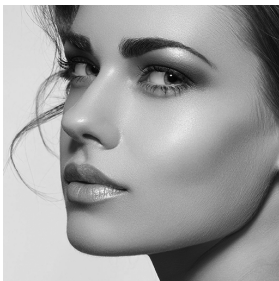
```
A = magic(3);  
B = ones(3);  
A(8,8) = 0; % zero-pad A to be 8-by-8  
B(8,8) = 0; % zero-pad B to be 8-by-8  
C = ifft2(fft2(A).*fft2(B));  
C = C(1:5,1:5); % extract the nonzero portion  
C = real(C) % remove imaginary part caused by roundoff error
```

$$C = \begin{bmatrix} 8.0000 & 9.0000 & 15.0000 & 7.0000 & 6.0000 \\ 11.0000 & 17.0000 & 30.0000 & 19.0000 & 13.0000 \\ 15.0000 & 30.0000 & 45.0000 & 30.0000 & 15.0000 \\ 7.0000 & 21.0000 & 30.0000 & 23.0000 & 9.0000 \\ 4.0000 & 13.0000 & 15.0000 & 11.0000 & 2.0000 \end{bmatrix}$$

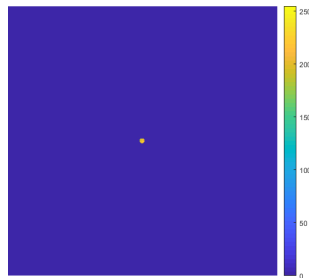
Poznámka: konvoluce přes spektra je v MATLABu rychlejší pro velké matice. Pro malé matice je rychlejší `conv2`, `filt2`.

Dolnoproustní filtrace, ořezání v ostrém kruhu, $r=5, 15, 50$

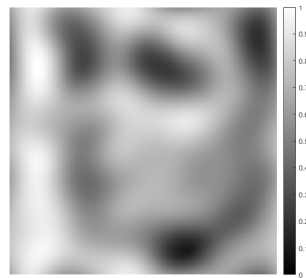
originál



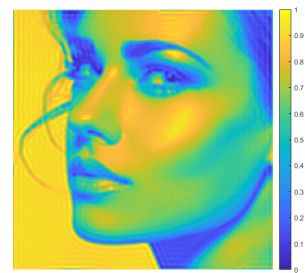
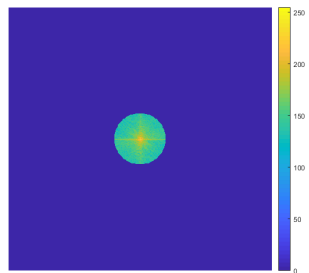
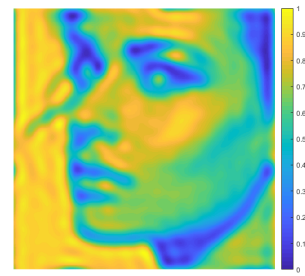
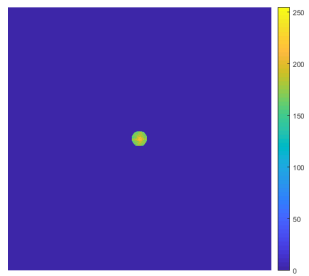
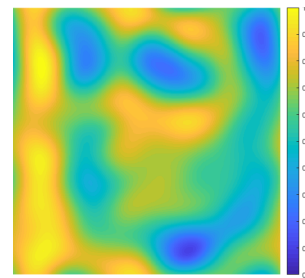
filtr



šedotónový výstup

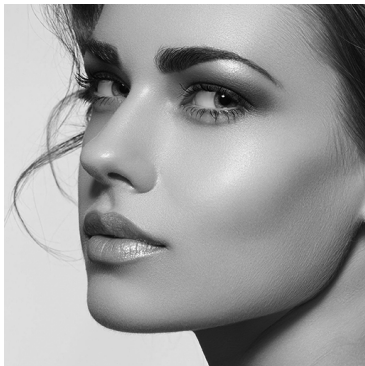


pseudobarva

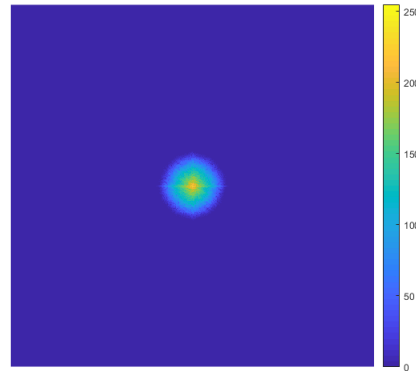


Dolnoproustní filtrace Gaussiánem, $\sigma = 10, 30$

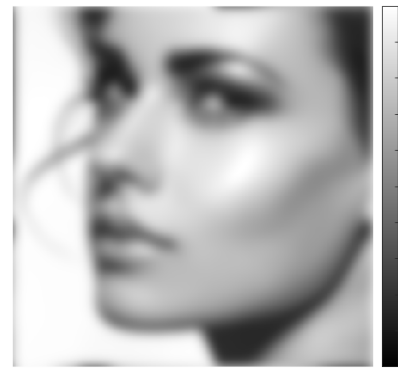
originál



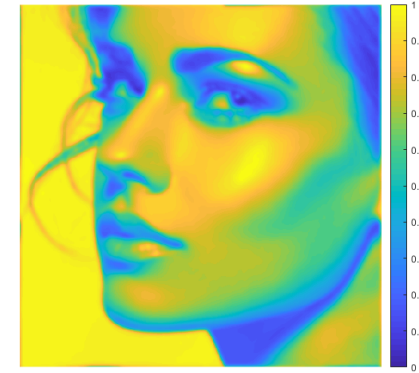
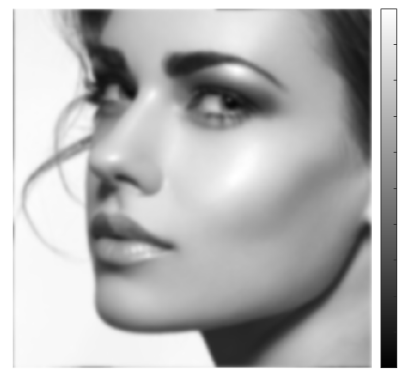
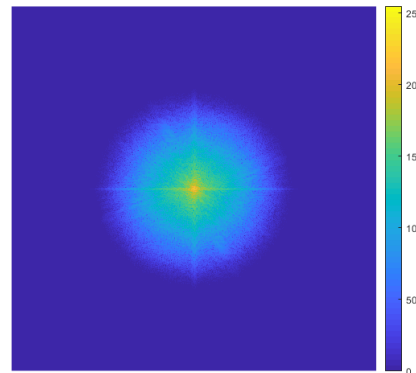
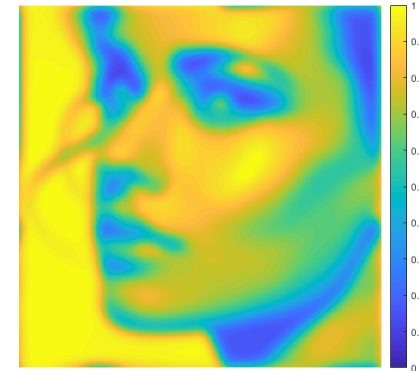
filtr



šedotónový výstup

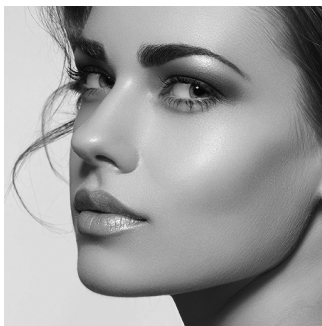


pseudobarva

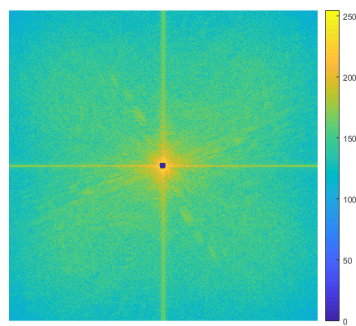


Hornopropustní filtrace, ořezání v ostrém kruhu, $r=5, 15, 50$

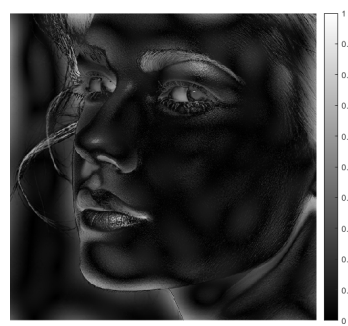
originál



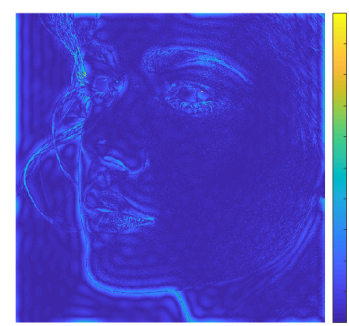
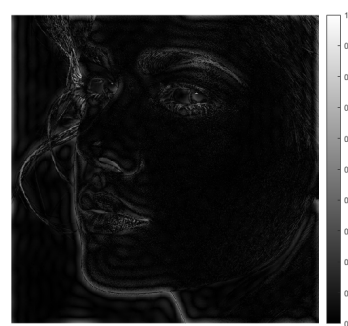
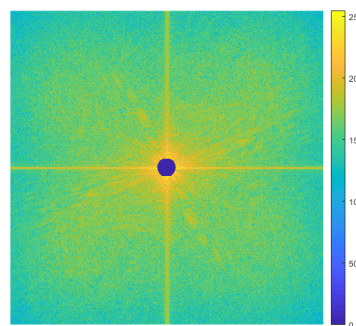
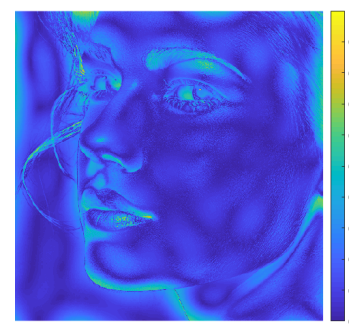
filtr



šedotónový výstup

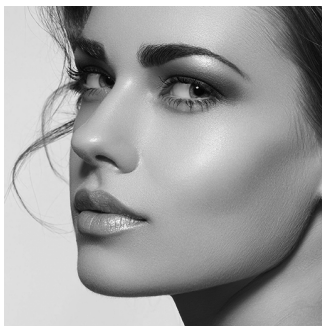


pseudobarva

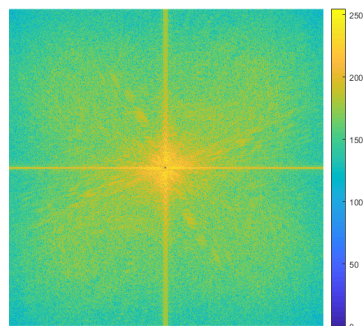


Hornopropustní filtrace Gaussiánem, $\sigma = 10, 30$

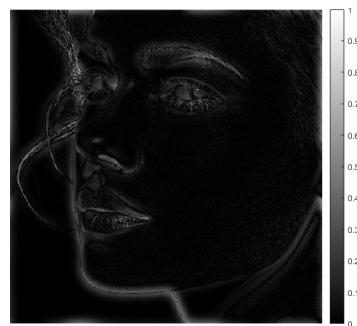
originál



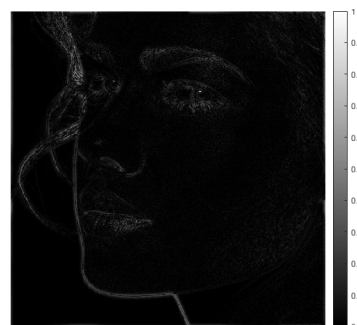
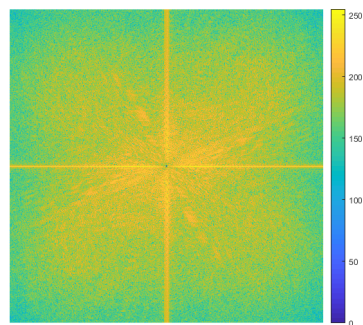
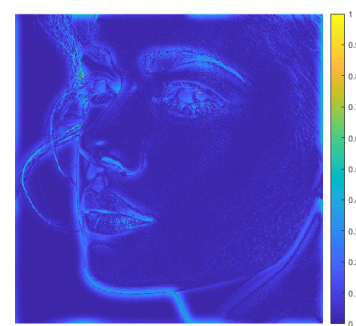
filtr



šedotónový výstup



pseudobarva

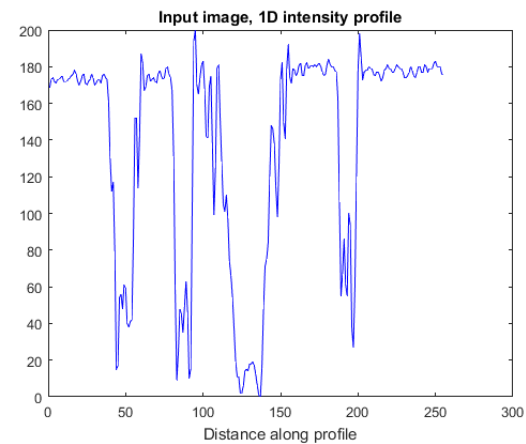
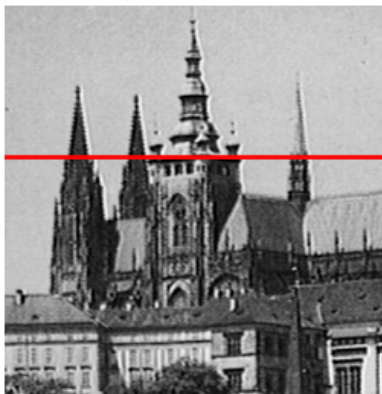


Příklad, fourierovská filtrace, gaussovský filtr

Input image



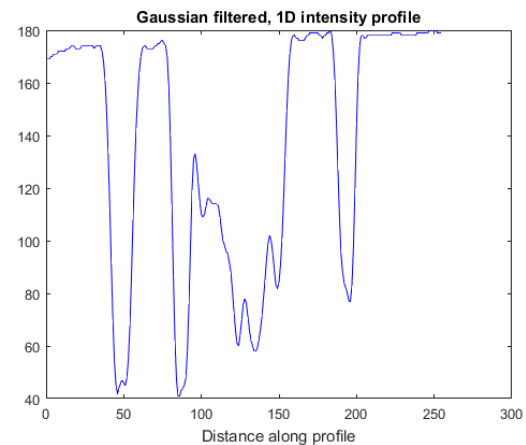
Input image, line for profile in red



Gaussian filtered



Gaussian filtered, line for profile in red

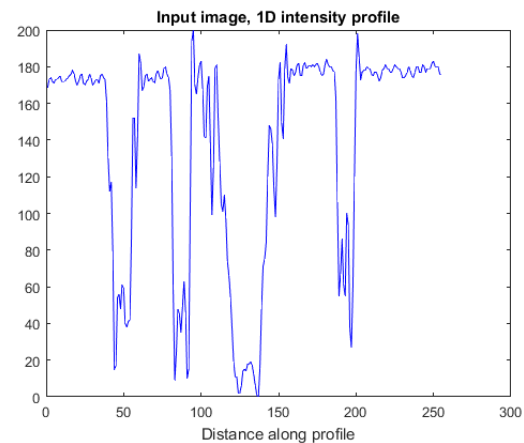
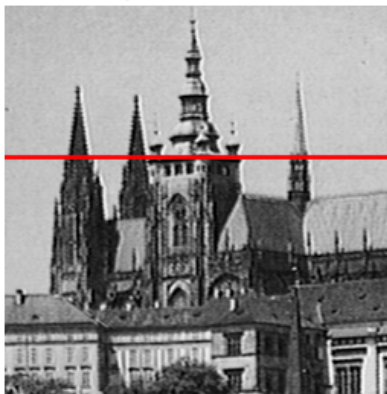


Příklad, fourierovská filtrace, Sobelův filtr

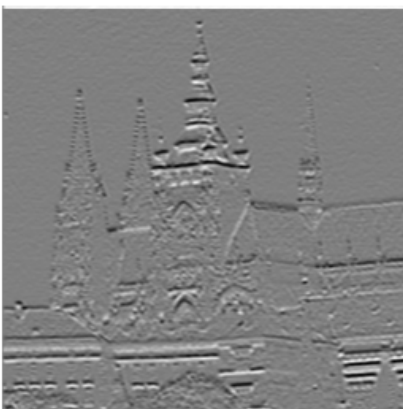
Input image



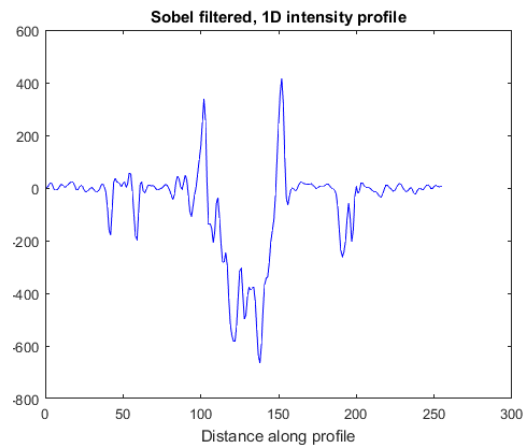
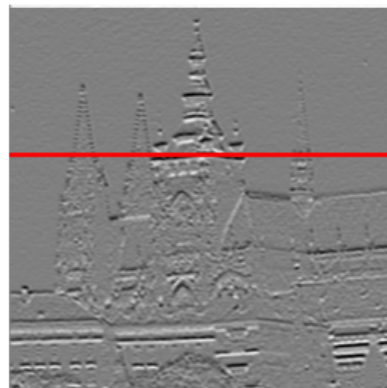
Input image, line for profile in red



Sobel filtered

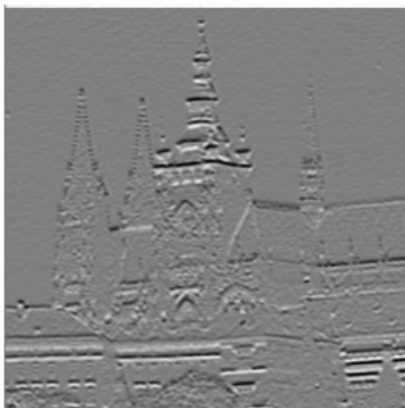


Sobel filtered, line for profile in red

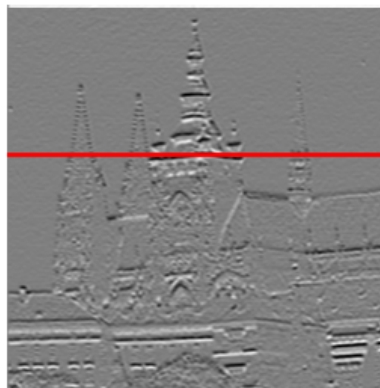


Příklad, fourierovská filtrace, Abs(Sobelův filtr)

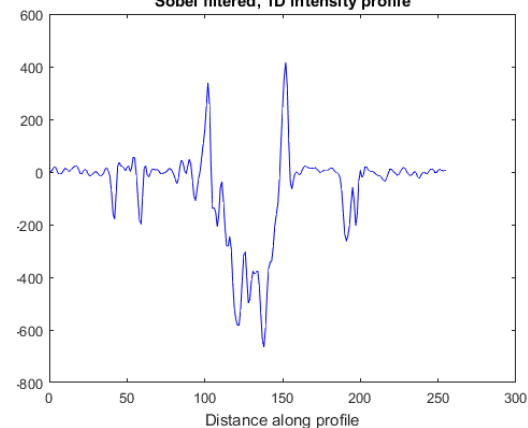
Sobel filtered



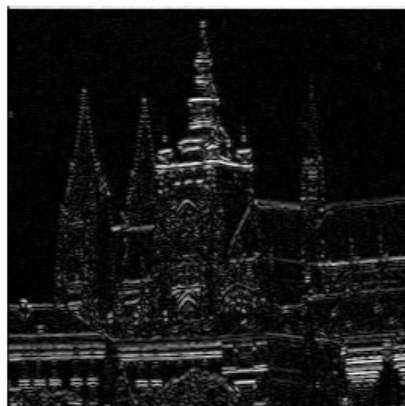
Sobel filtered, line for profile in red



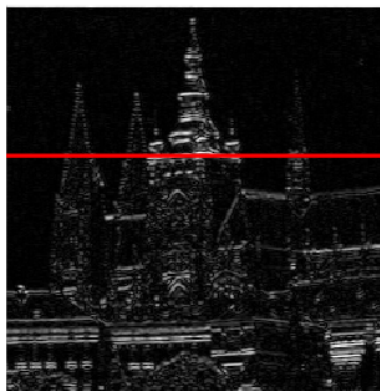
Sobel filtered, 1D intensity profile



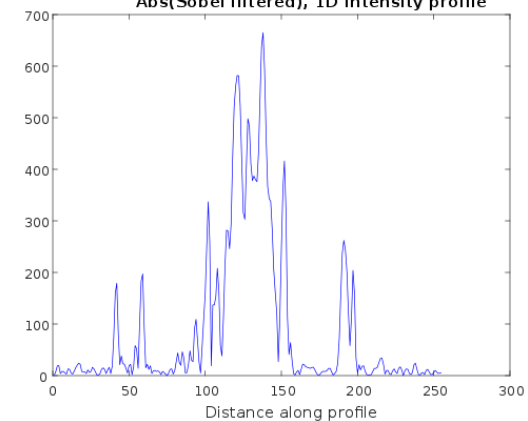
Abs(Sobel filtered)



Abs(Sobel filtered)



Abs(Sobel filtered), 1D intensity profile

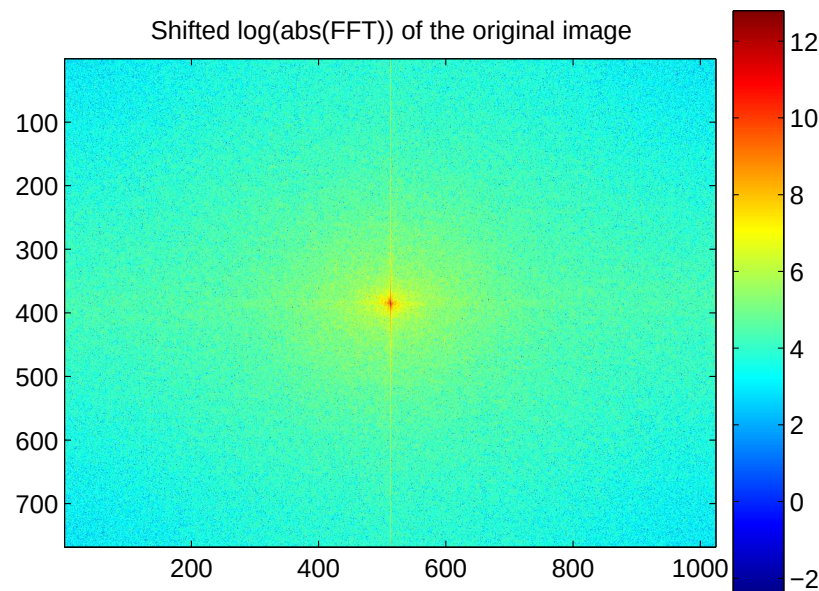


Dolní propust, Butterworth (1)

Filtr, který má nejméně zvlněné frekvenční spektrum a konverguje k nule u maximální frekvence (S. Buttherworth, 1930).

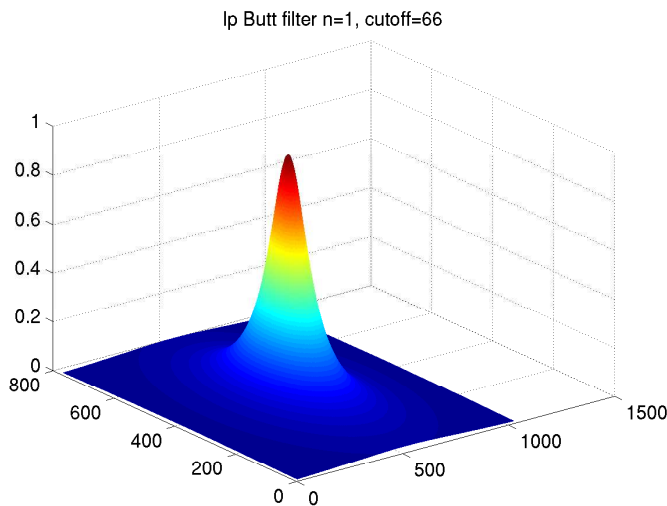


Výchozí obrázek

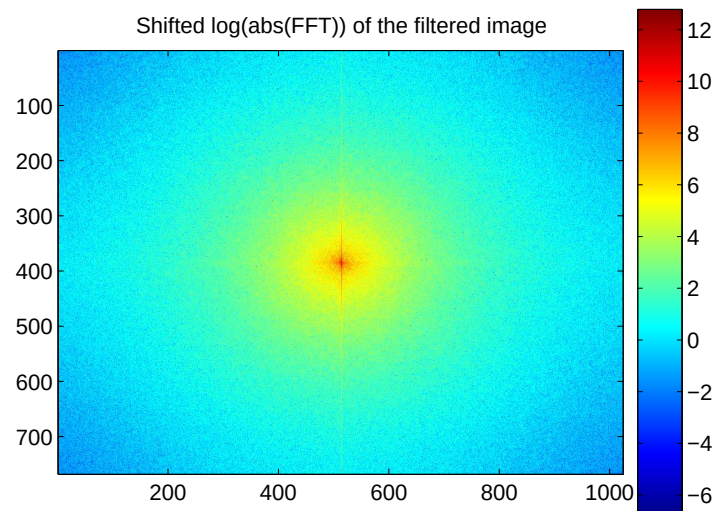


Frekvenční spektrum

Dolní propust, Butterworth (2)



Butterworthův dolnoproř. filtr



FFT filtrovaného obrázku

$$H(u, v) = \frac{1}{1 + \left(\frac{D(u, v)}{D_0}\right)^{\frac{2}{n}}}, \text{ kde } D(u, v) = \sqrt{u^2 + v^2}. \text{ Číslo } n \text{ je stupeň filtru.}$$

D_0 je frekvence filtru odpovídající poklesu amplitudy o 3dB.

Dolní propust, Butterworth (3)



Výchozí obrázek



Filtrovaný obrázek

Homomorfický filtr (1)

Cíl: normalizovat současně jas přes celý obrázek a zvýšit kontrast.

Vychází se z předpokladů, že:

- ♦ **osvětlení** i se mění v obrázku pomalu (malé frekvence),
- ♦ **odrazivost** r se mění rychle, protože scéna bývá různorodá,
- ♦ obrázek lze **rozložit v každém pixlu na součin dvou složek** – osvětlení i a odrazivosti r :
$$f(x, y) = i(x, y) r(x, y).$$

Hlavní myšlenka: použít funkci logaritmus pro oddělení komponenty osvětlení a komponenty odrazivosti.

Homomorfický filtr (2)

$$z(x, y) = \ln f(x, y) = \ln i(x, y) + \ln r(x, y)$$

$$\text{Fourierovo spektrum } Z(u, v) = I(u, v) + R(u, v)$$

Filtrování ve frekvenční oblasti $S(u, v) = H(u, v)Z(u, v) = H(u, v)I(u, v) + H(u, v)R(u, v)$

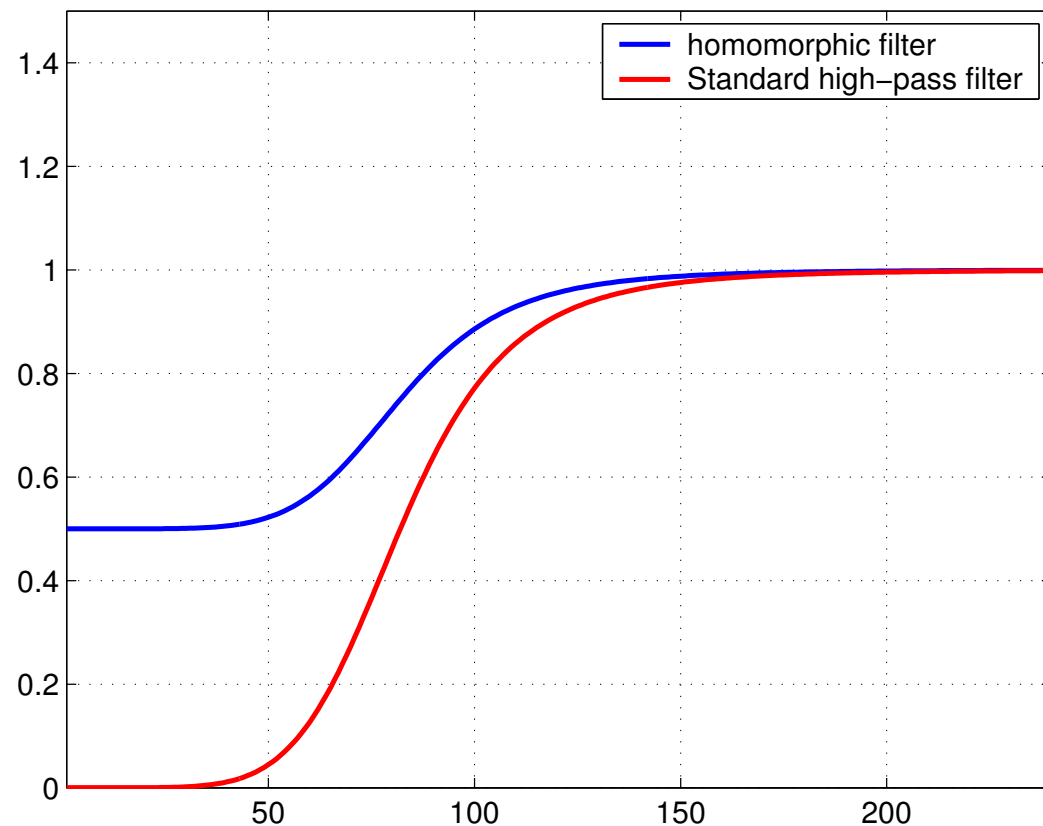
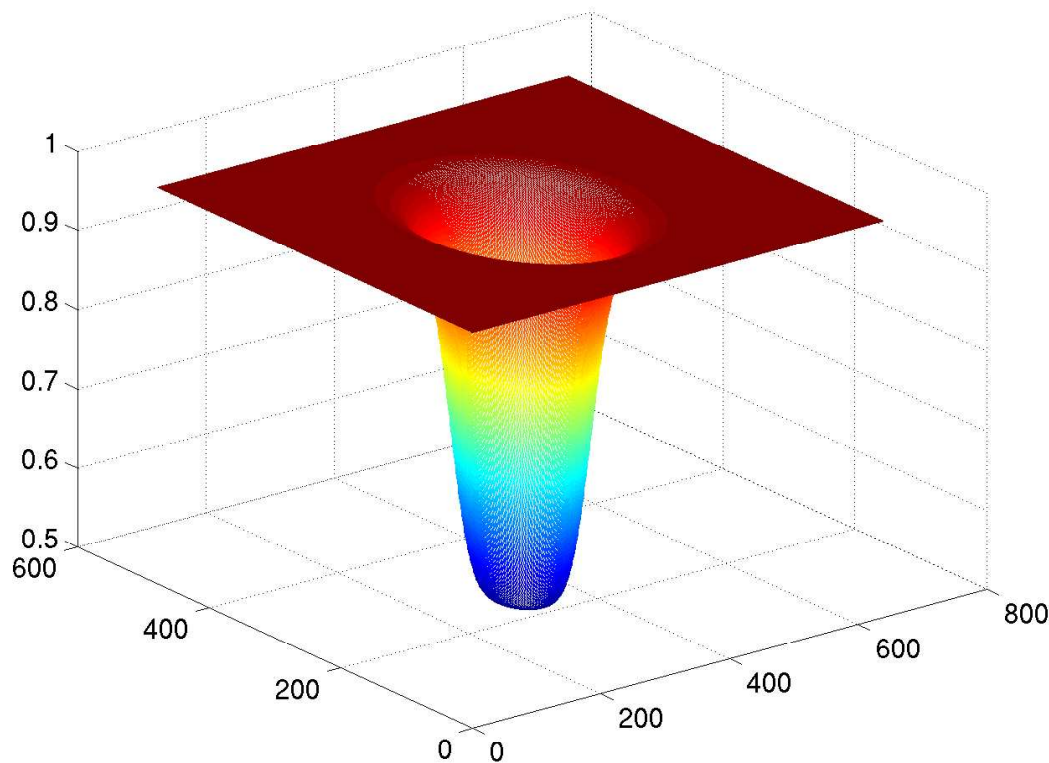
převod zpět do prostorových souřadnic $s(x, y) = \mathcal{F}^{-1}\{S(u, v)\}$ a návrat z $\ln$

$$g(x, y) = \exp(s(x, y))$$

Výsledkem je potlačení změn v osvětlení scény a zlepšení odrazivostní složky.

Homomorické filtry

Homomorphic filter made by adaptation of Buttherworth highpass



Připomínka: Filtr je použit na $Z(u, v)$ a nikoliv na původní spektrum $F(u, v)$!

Výsledky homomorfické filtrace



Výchozí obrázek



Filtrovaný obrázek

Návrh 2D FIR filtrů

- ◆ Dvojměrné IIR filtry se nepoužívají kvůli nestabilitě (není zaručena kauzalita).
- ◆ FIR filtry se snadno reprezentují jako matice koeficientů. Snadná implementace.
- ◆ 2D FIR filtry jsou přirozeným zobecněním 1D FIR filtrů.
- ◆ FIR filtry lze navrhnout, aby měly lineární fázi, což omezuje zkreslení.
- ◆ Používají se **tři návrhové metody**:
 1. **Metoda transformace frekvence** převádí 1D filtr na 2D.
 2. **Metoda vzorkování frekvence** vytváří filtr podle požadované frekvenční odezvy.
 3. **Metoda okna** skládá filtr z ideální impulsní charakteristiky a vyhlazovacího okna.

Metoda transformace frekvence

Lze využít metody pro návrh 1D filtrů. Převod na středově souměrný 2D filtr. Dobrá metoda.

Příklad v MATLABU (Parks-McClellanův optimální návrh):

```
b = remez(10,[0 0.4 0.6 1],[1 1 0 0]);
```

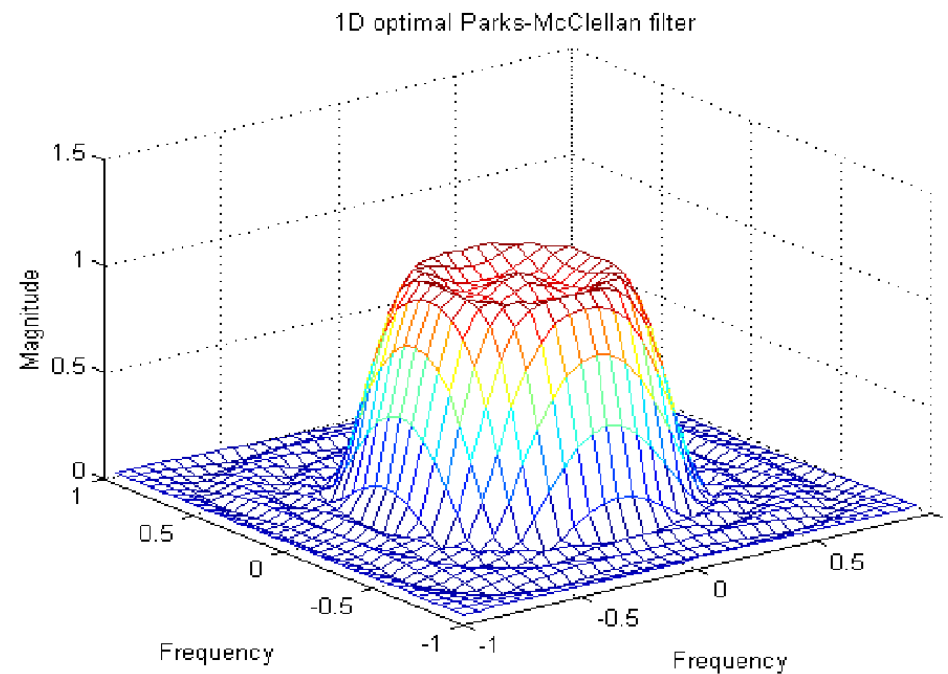
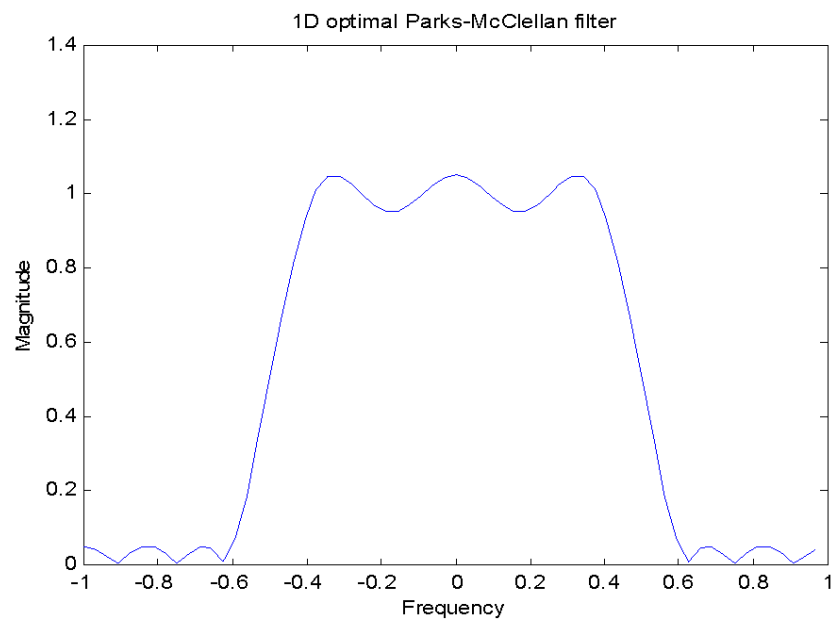
```
h = ftrans2(b);
```

```
[H,w] = freqz(b,1,64,'whole');
```

```
colormap(jet(64))
```

```
plot(w/pi-1,fftshift(abs(H))) figure, freqz2(h,[32 32])
```


2D Parks-McClellan Filtr pokračování příkladu



Metoda vzorkování frekvence

Vytváří filtr na základě zadané frekvenční odezvy ve formě matice, který danými body prochází. Mimo vzorkovací body je chování libovolné, obvykle zakmitávání.

Příklad v MATLABU (návrh 11×11 filtru)

```
Hd = zeros(11,11); Hd(4:8,4:8) = 1;
```

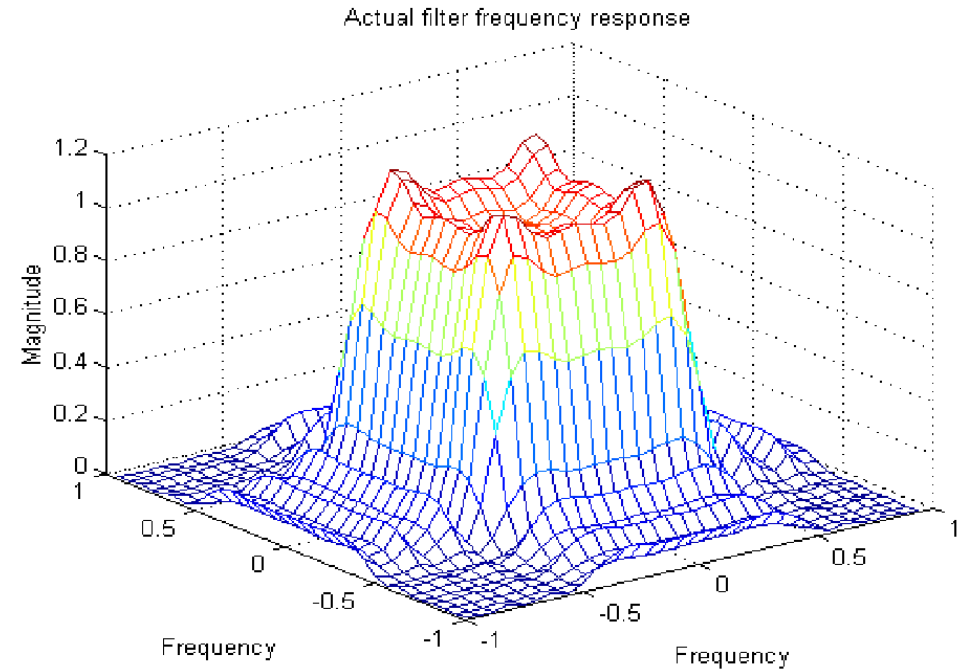
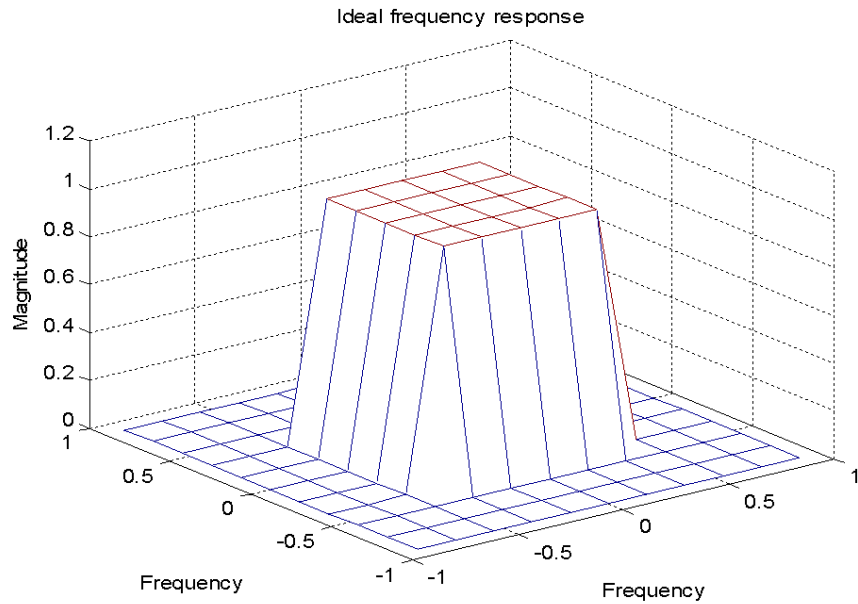
```
[f1,f2] = freqspace(11,'meshgrid');
```

```
mesh(f1,f2,Hd), axis([-1 1 -1 1 0 1.2]), colormap(jet(64))
```

```
h = fsamp2(Hd);
```

```
figure, freqz2(h,[32 32]), axis([-1 1 -1 1 0 1.2])
```

Metoda vzorkování frekvence pokračování příkladu



Metoda okna

Ideální odezva filtru se vyhladí koeficienty v okně. Aproximuje se ideální filtr.

Většinou poskytuje lepší výsledky než metoda vzorkování frekvence.

```
Hd = zeros(11,11); Hd(4:8,4:8) = 1;
```

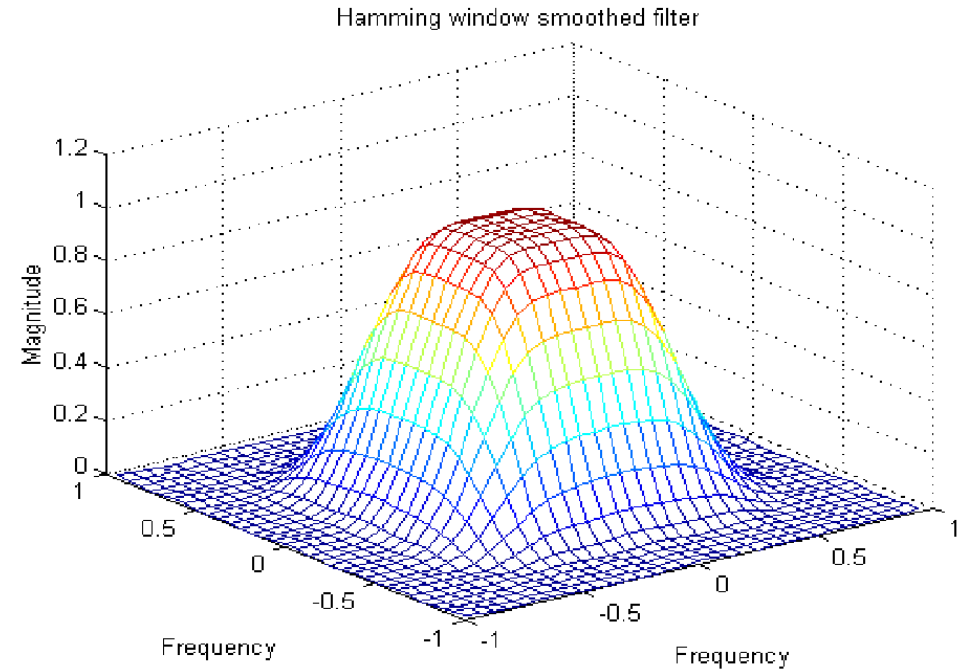
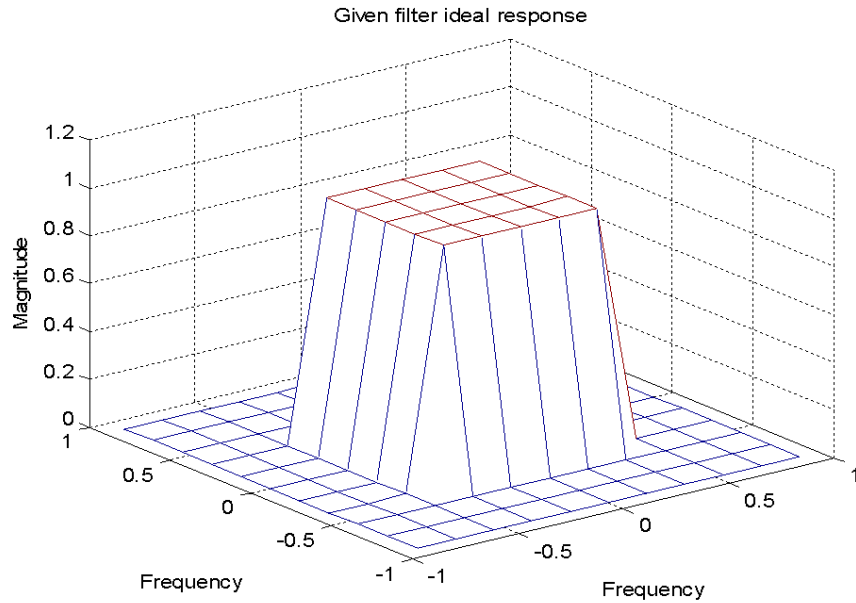
```
[f1,f2] = freqspace(11,'meshgrid');
```

```
mesh(f1,f2,Hd), axis([-1 1 -1 1 0 1.2]), colormap(jet(64))
```

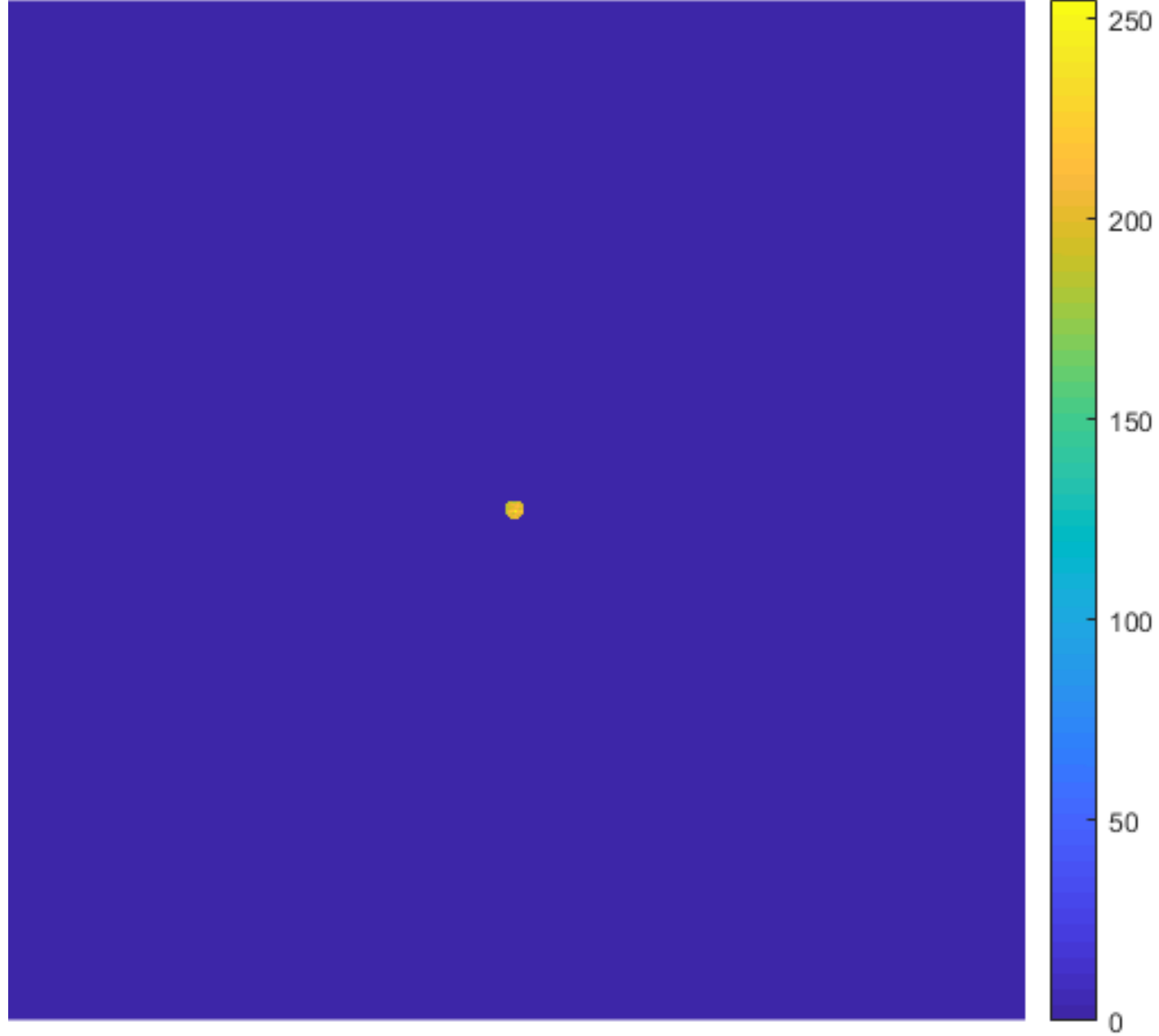
```
h = fwind1(Hd,hamming(11));
```

```
figure, freqz2(h,[32 32]), axis([-1 1 -1 1 0 1.2])
```

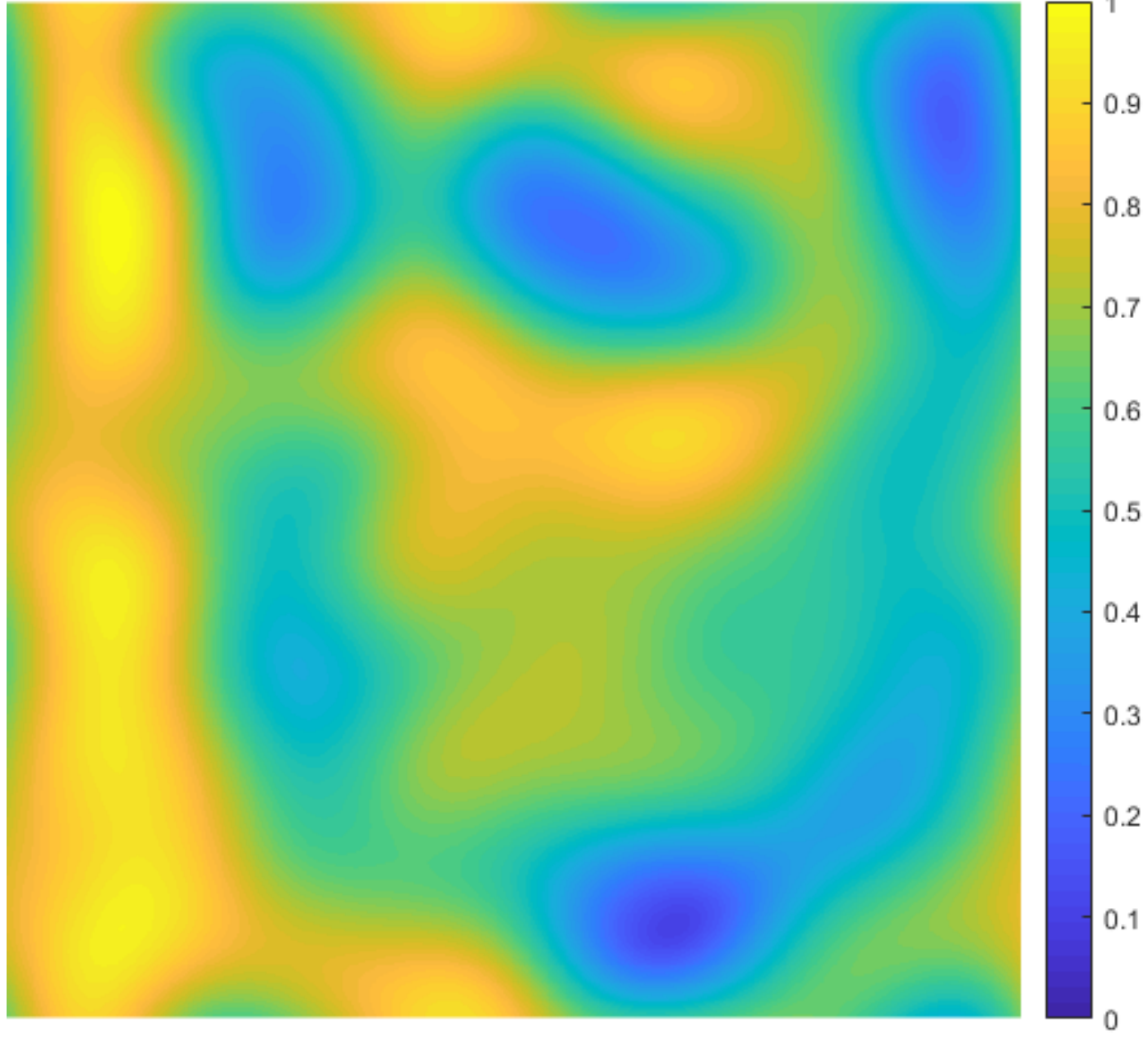
Metoda okna pokračování příkladu

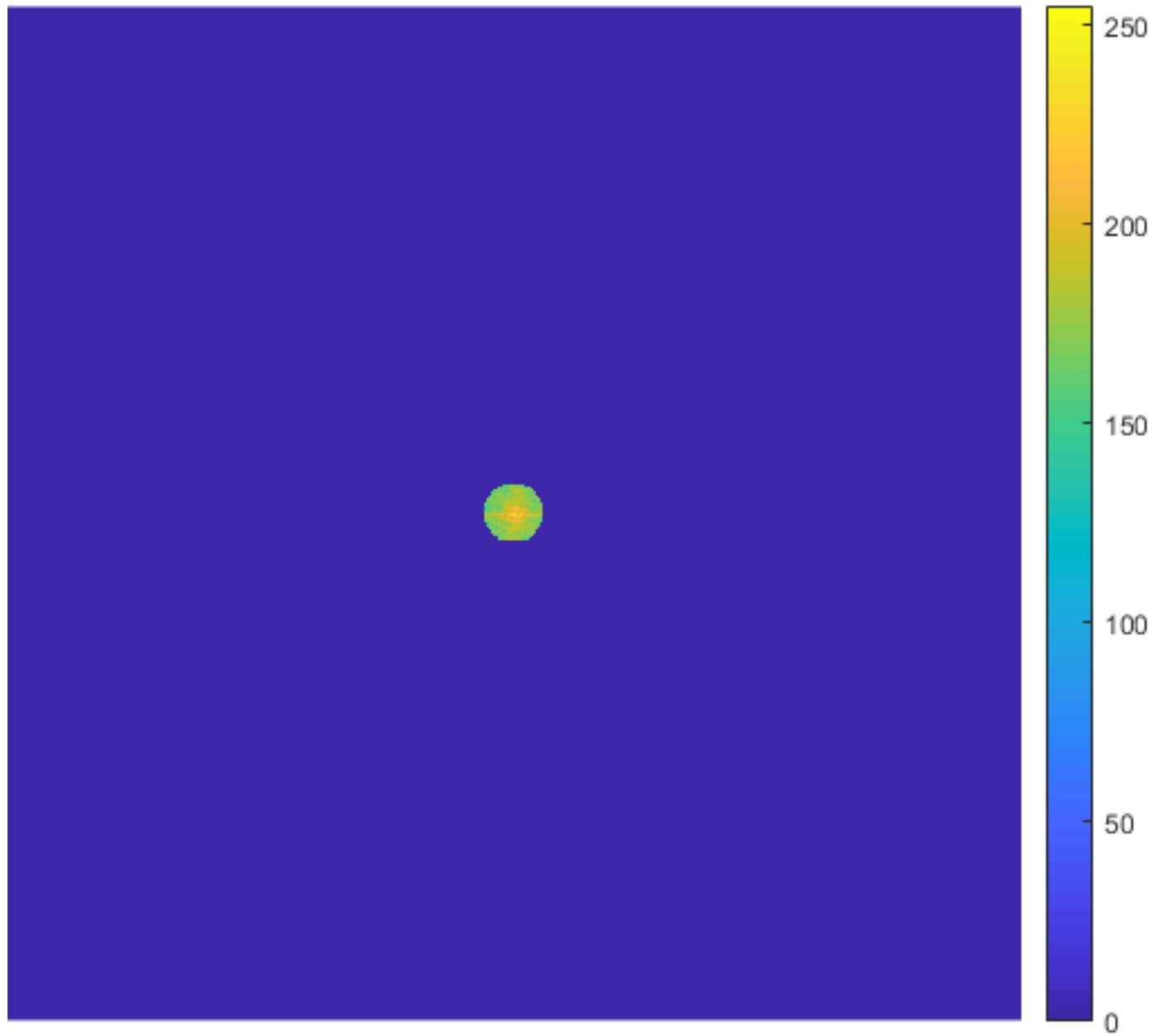


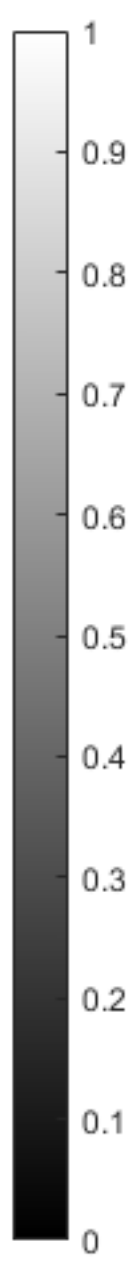


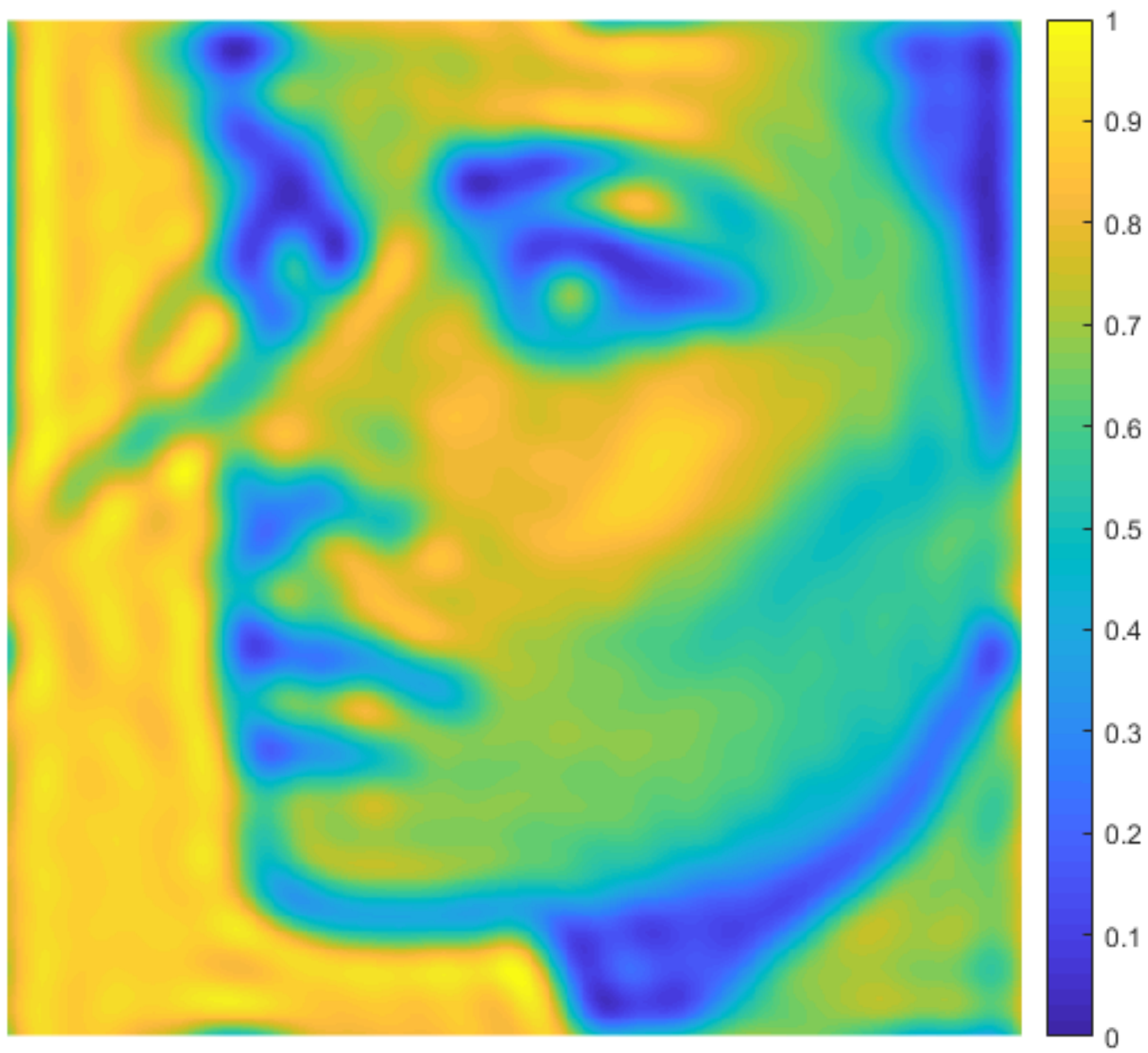


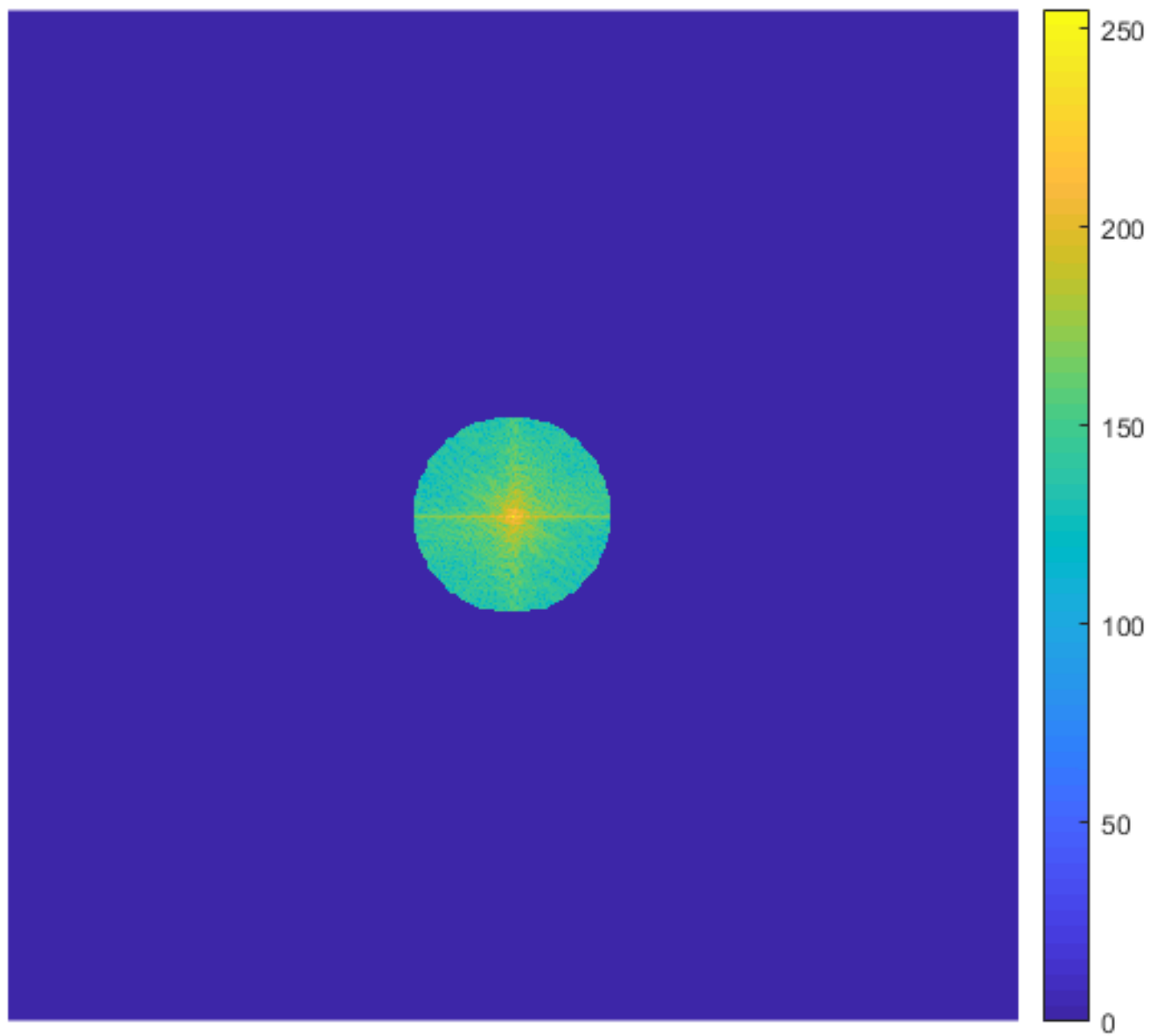




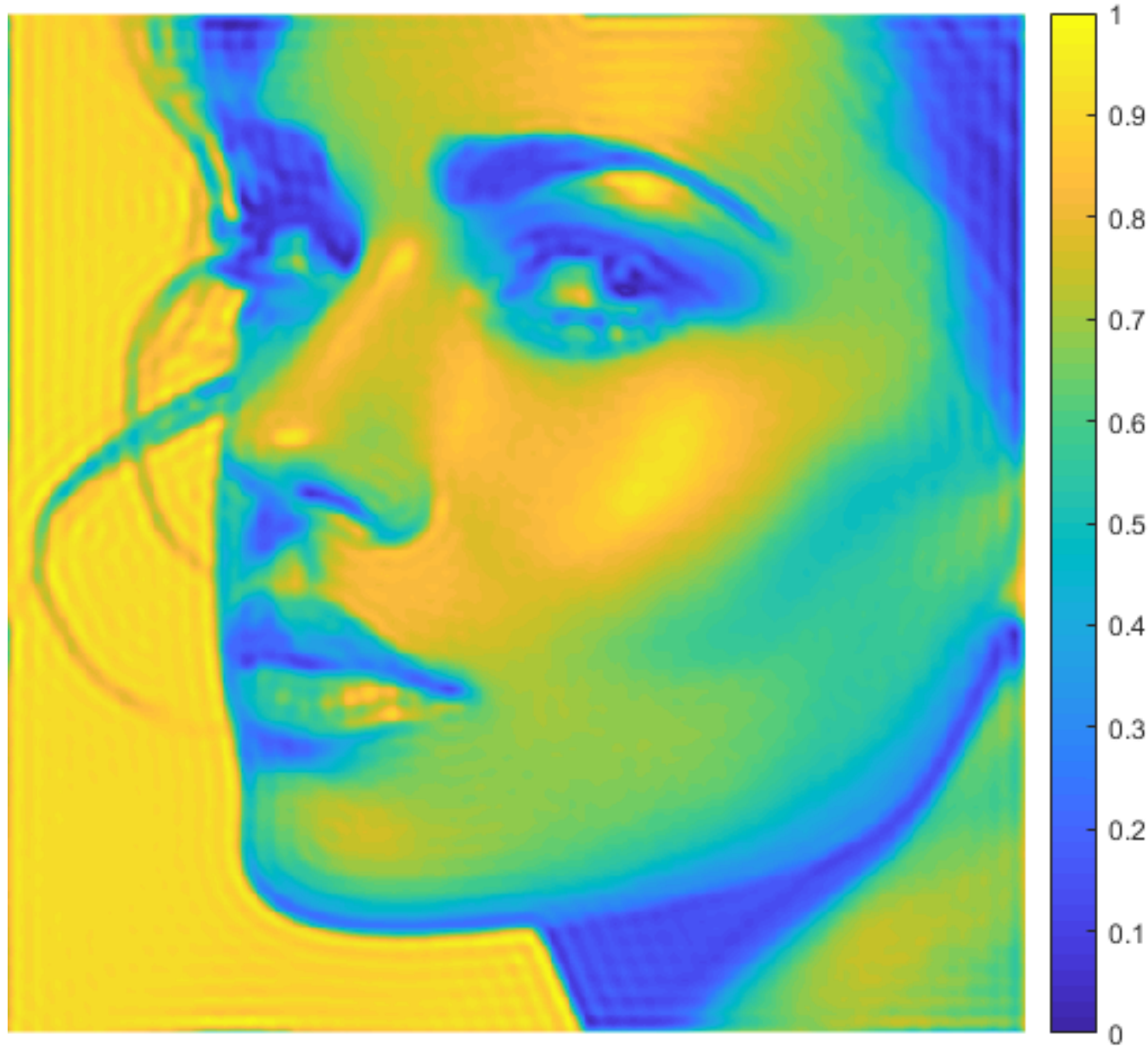




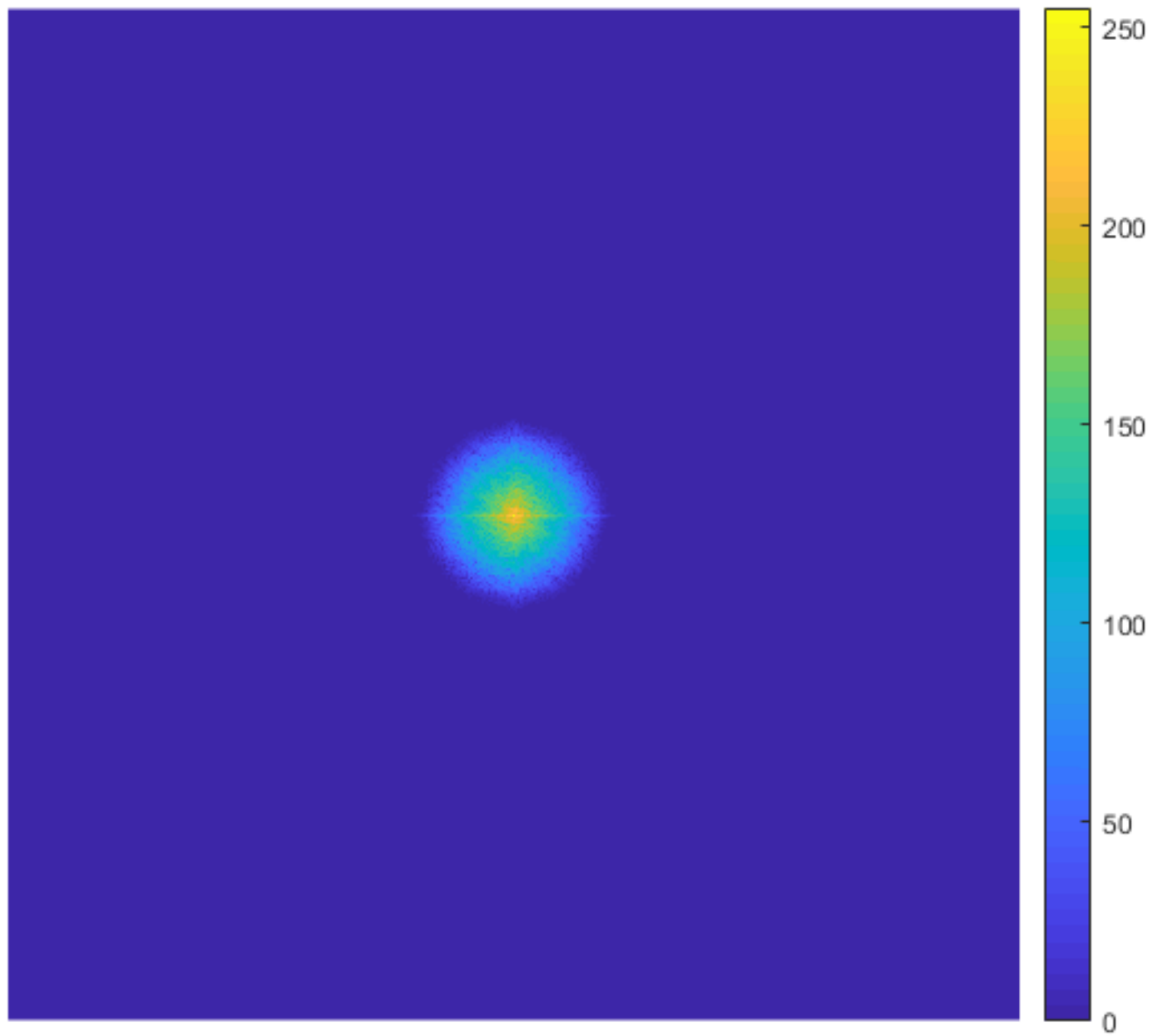


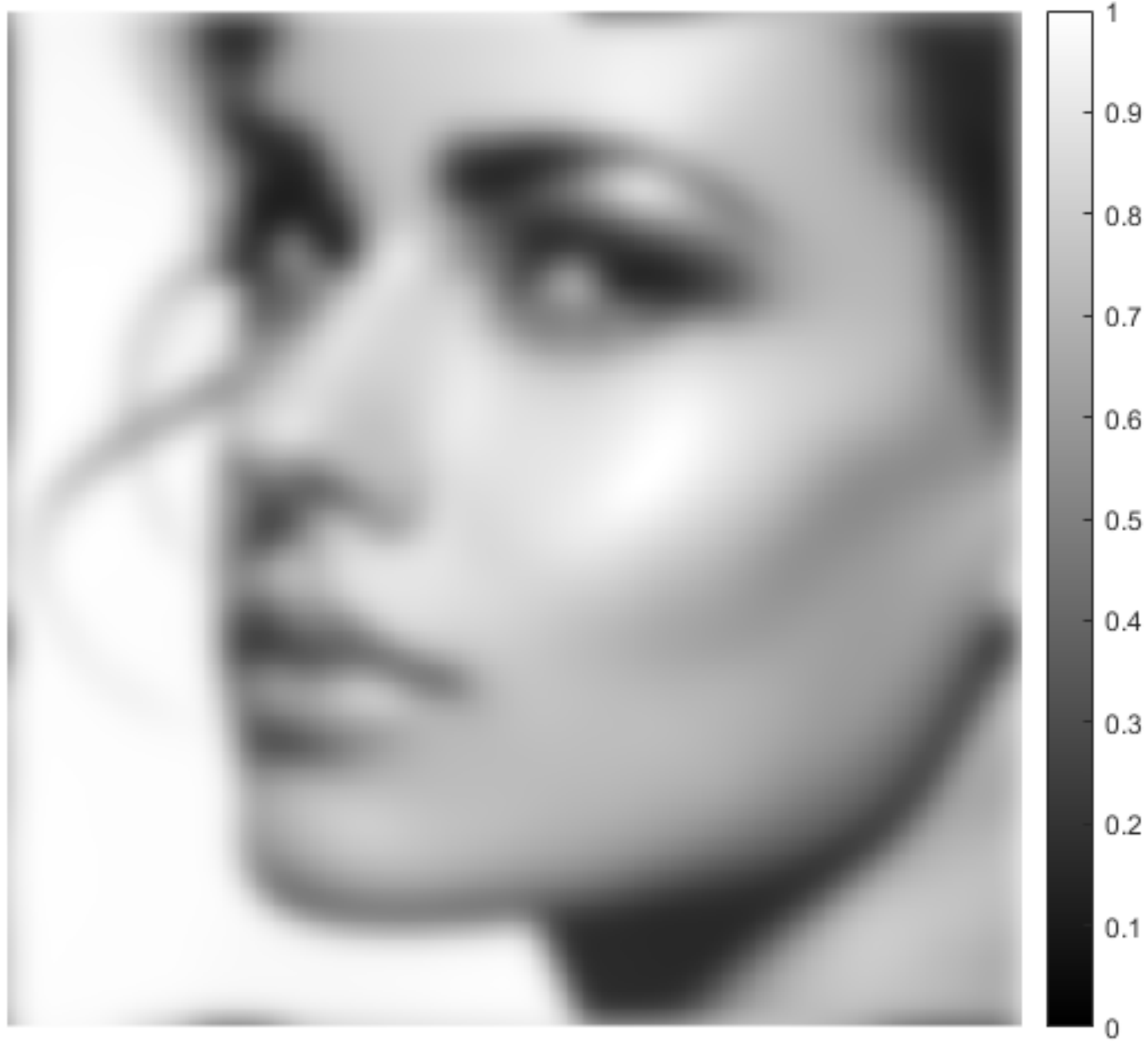


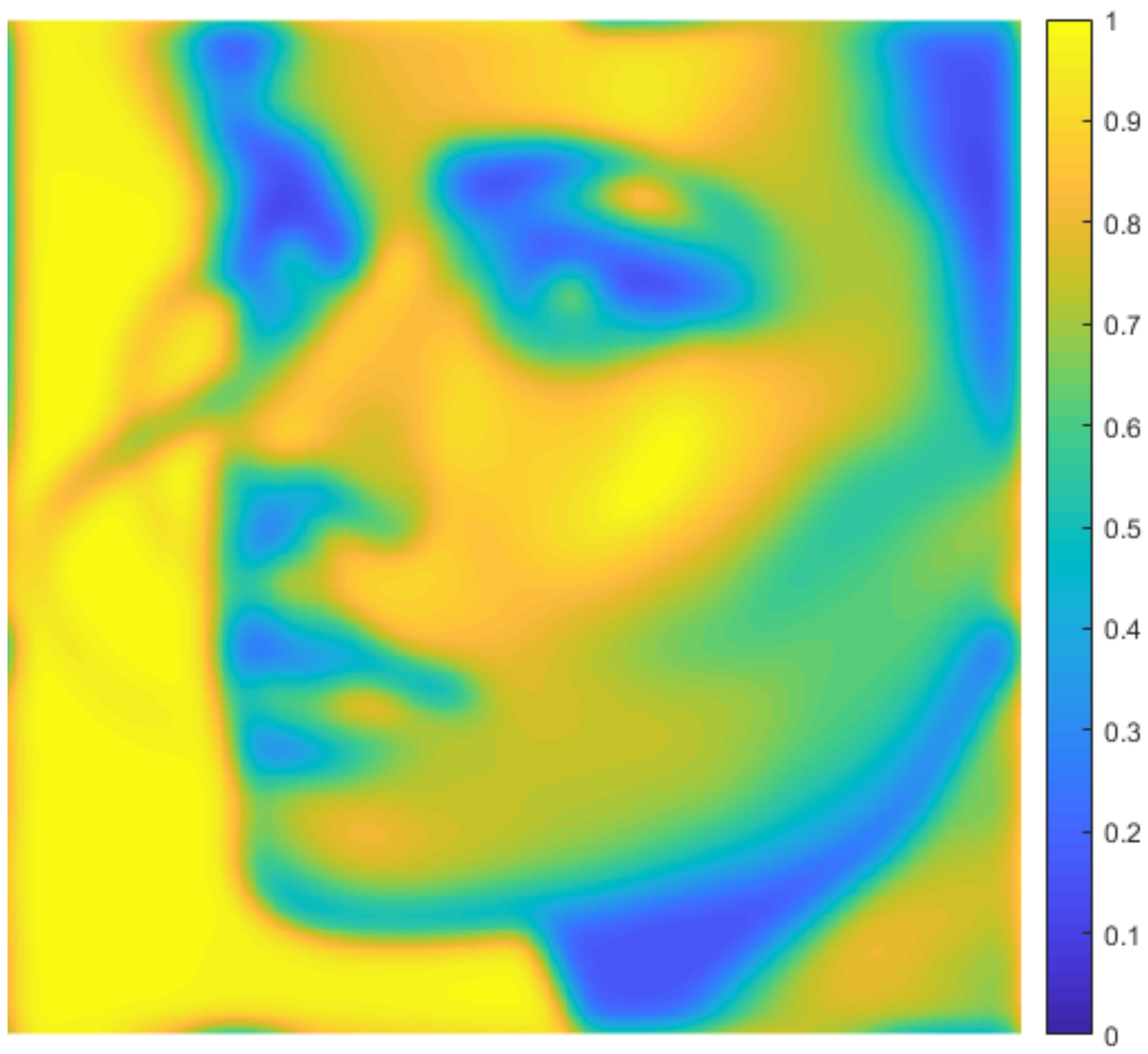


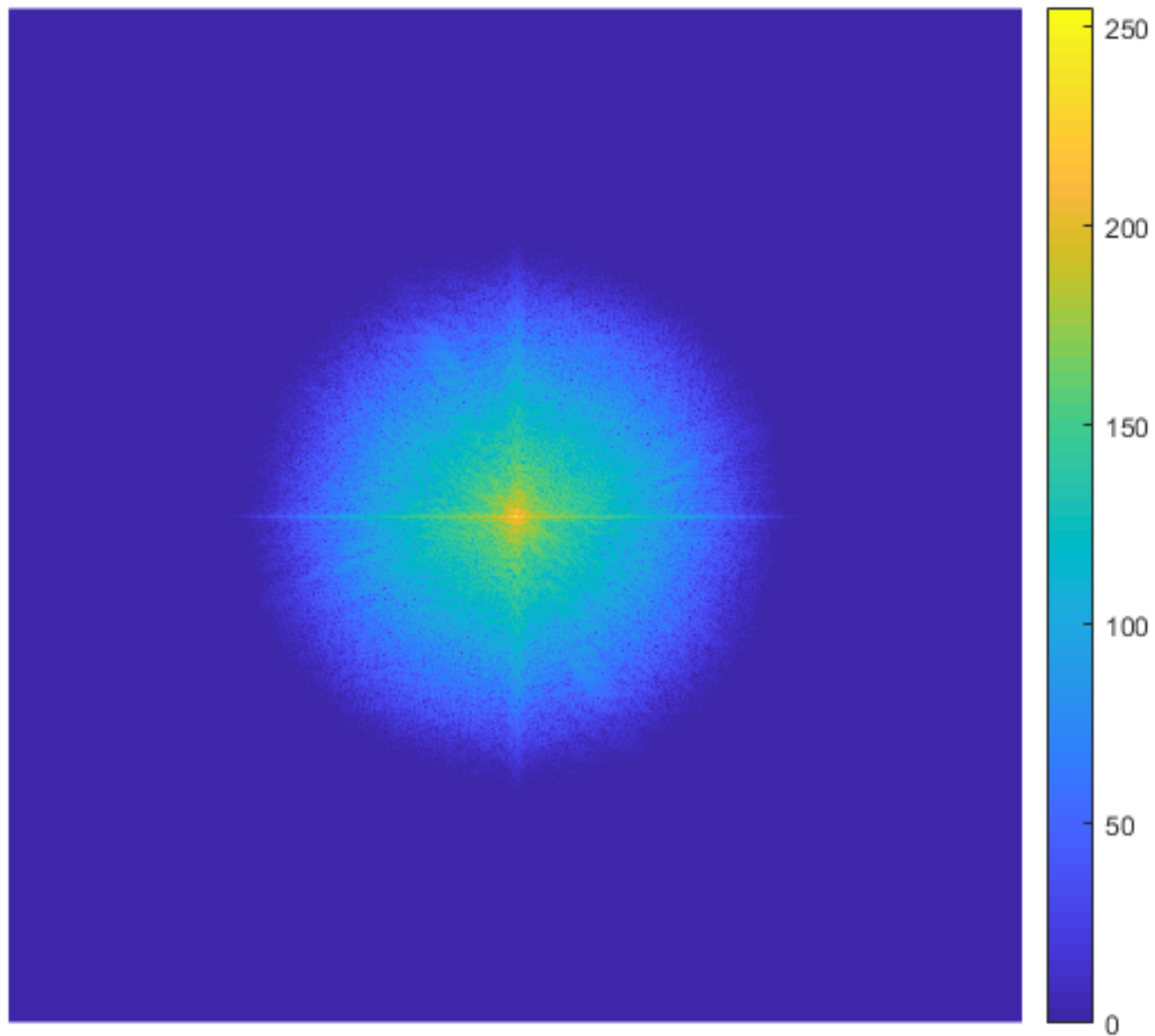


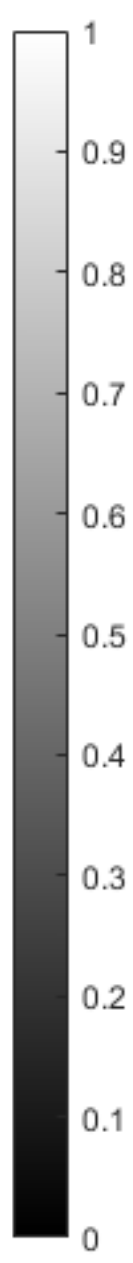
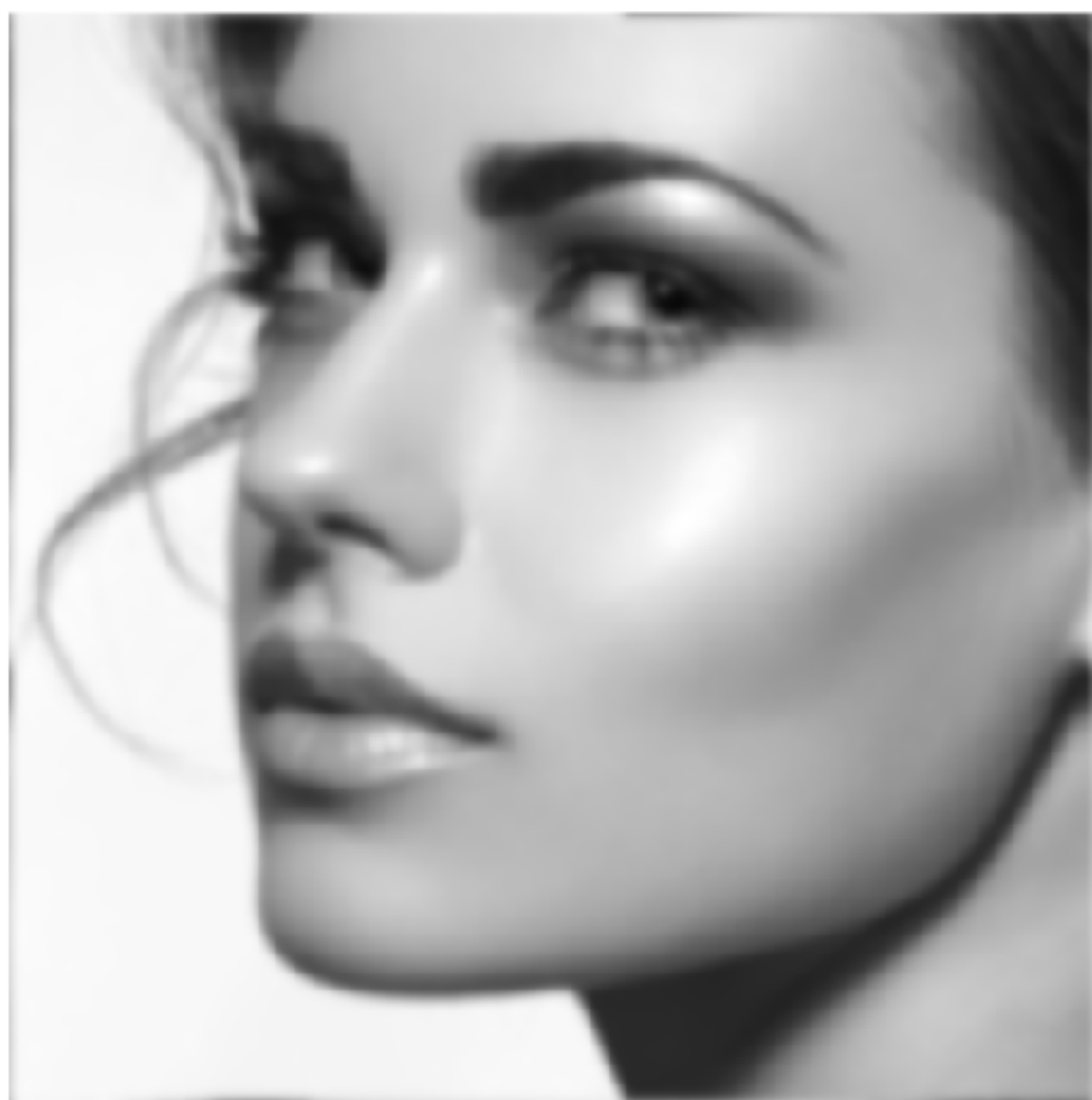


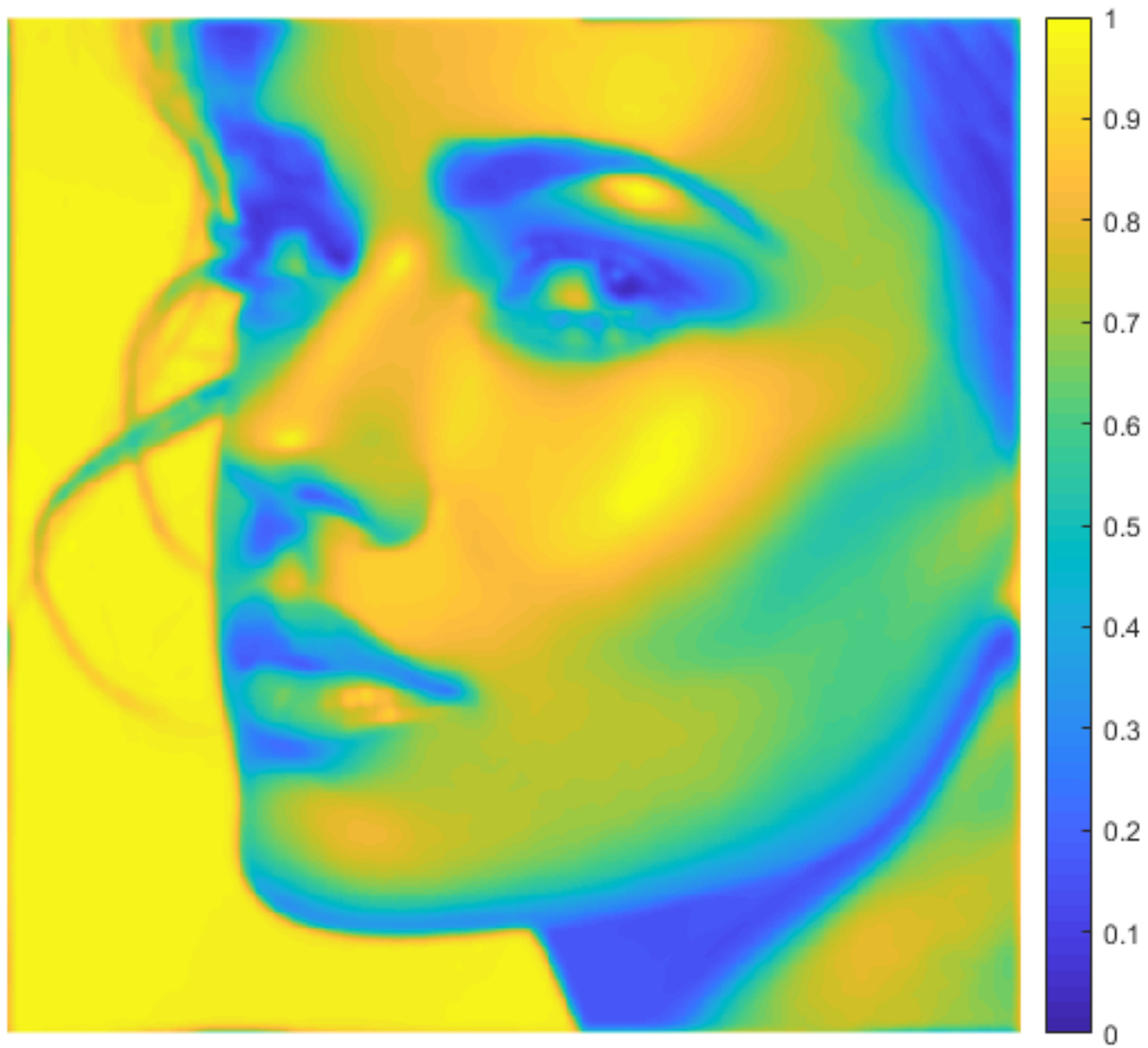




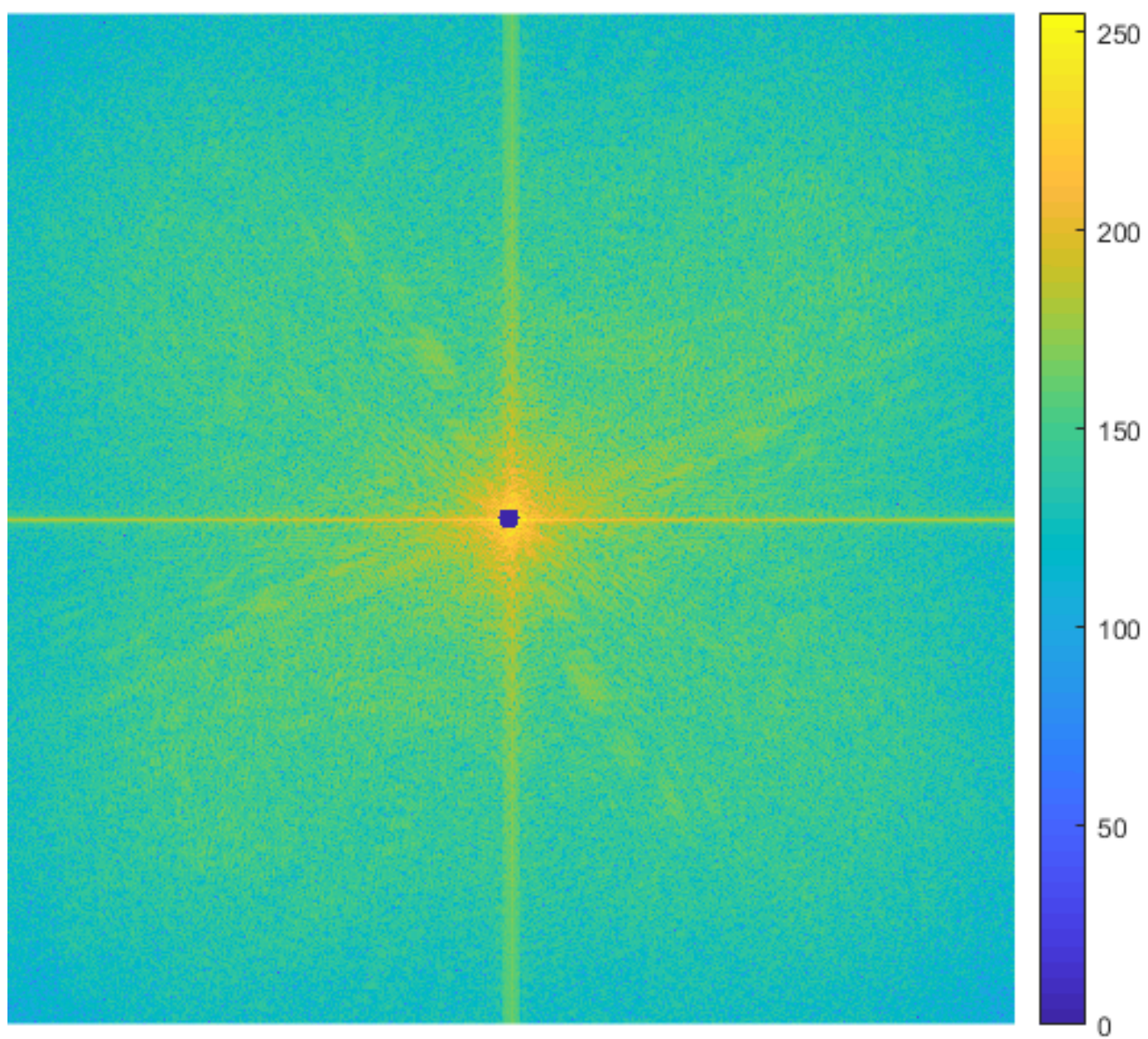


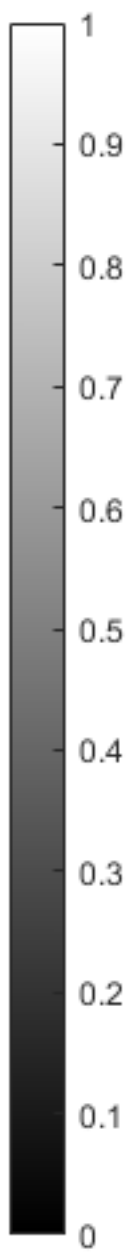
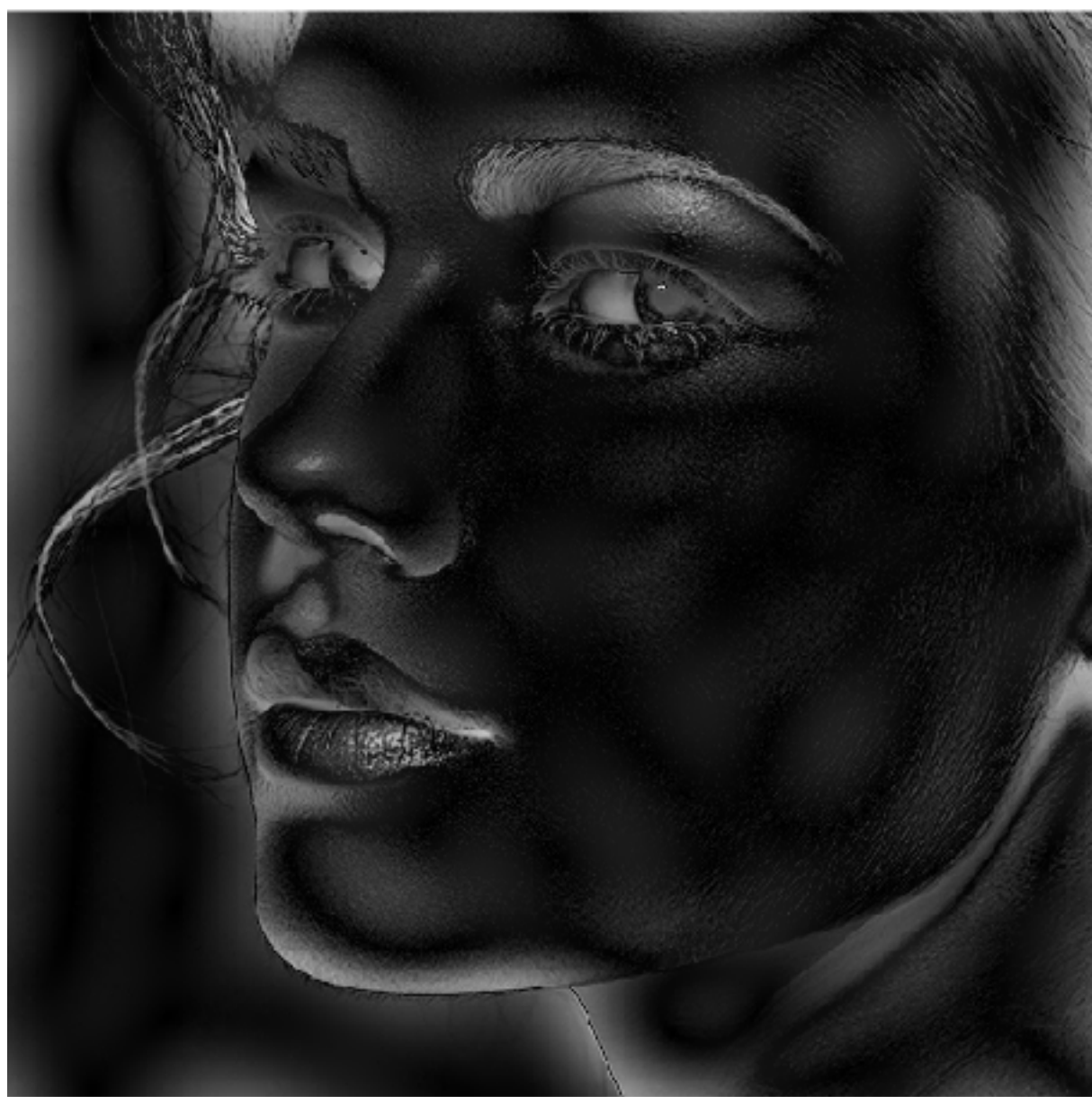


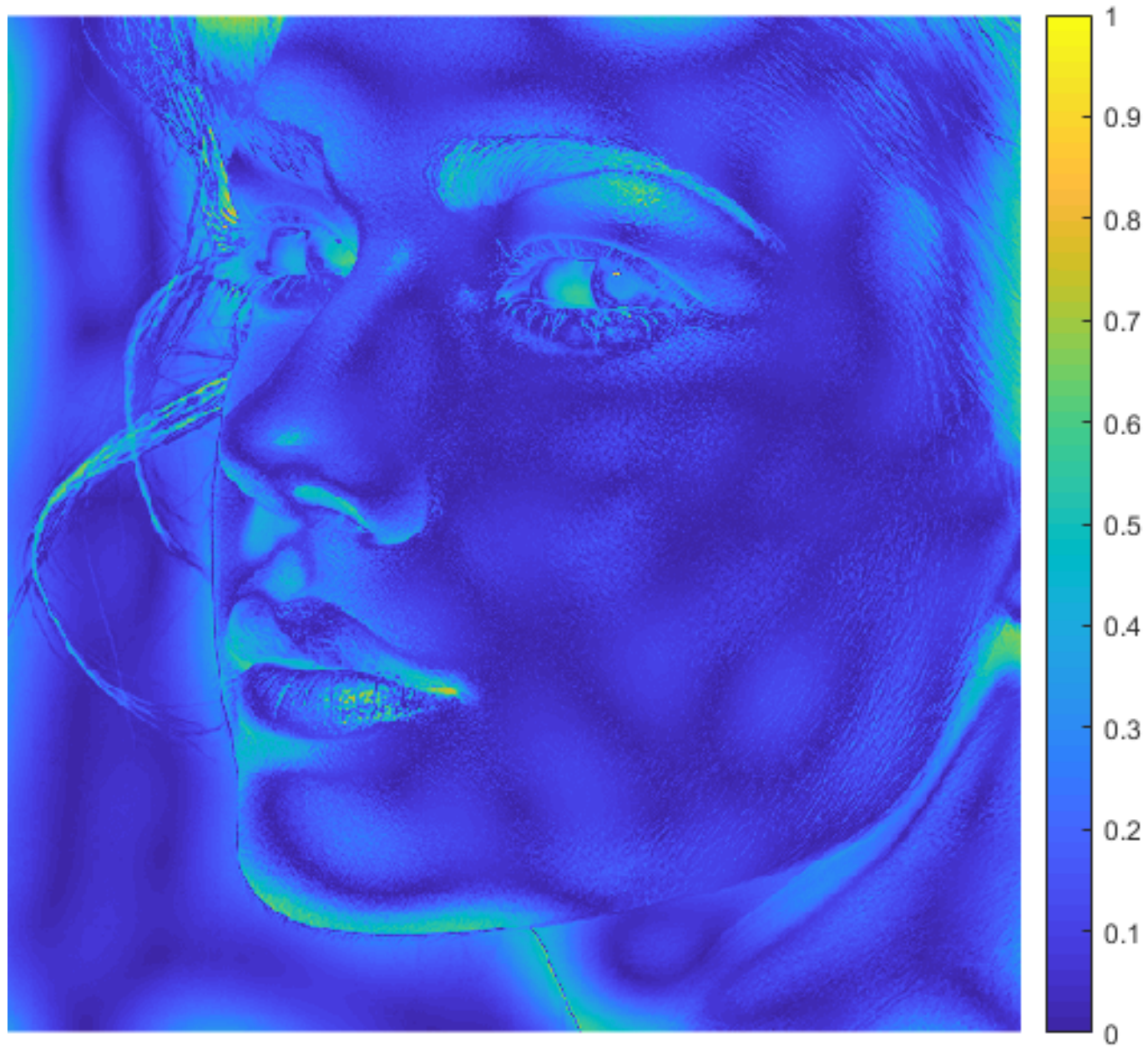


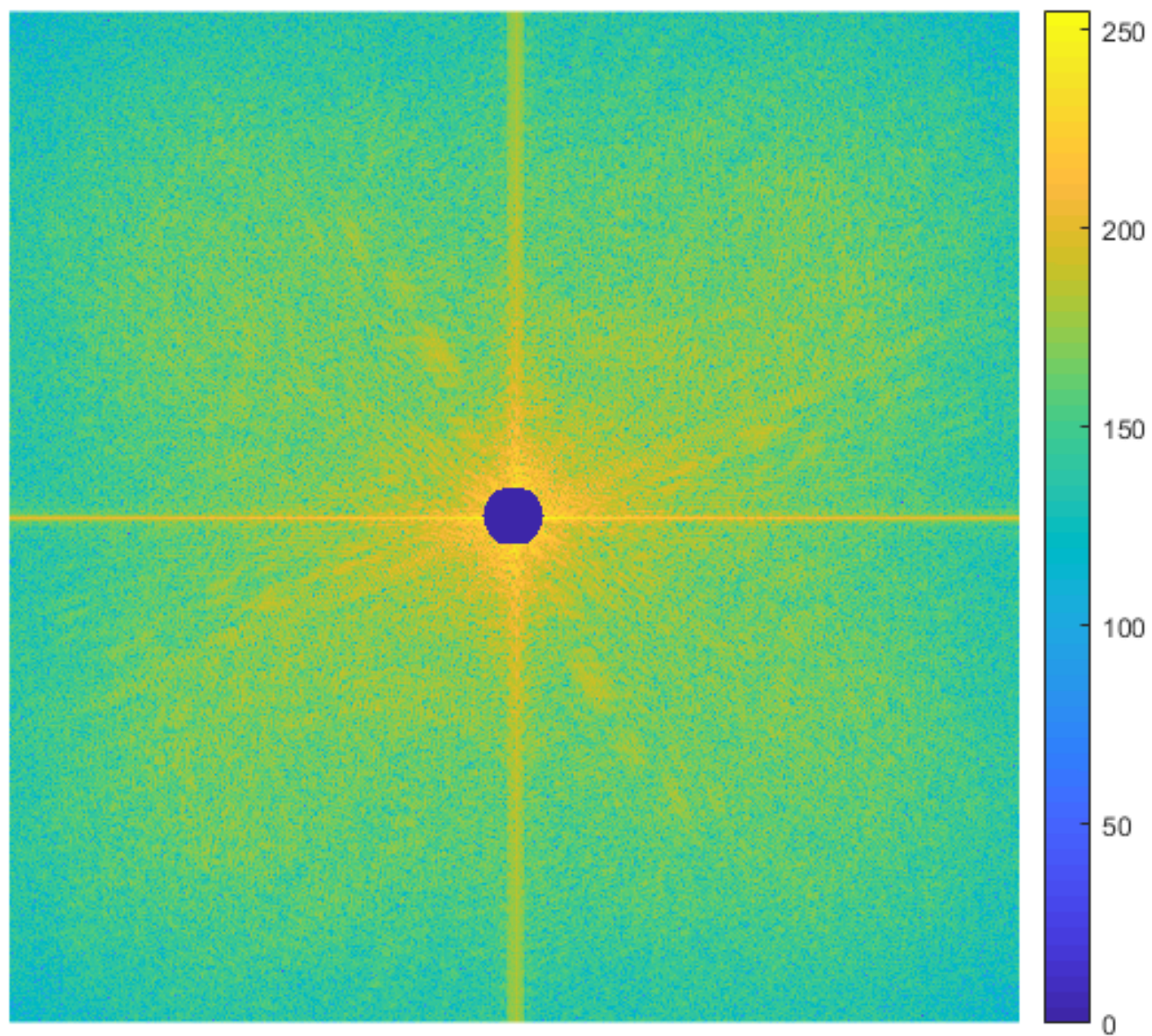


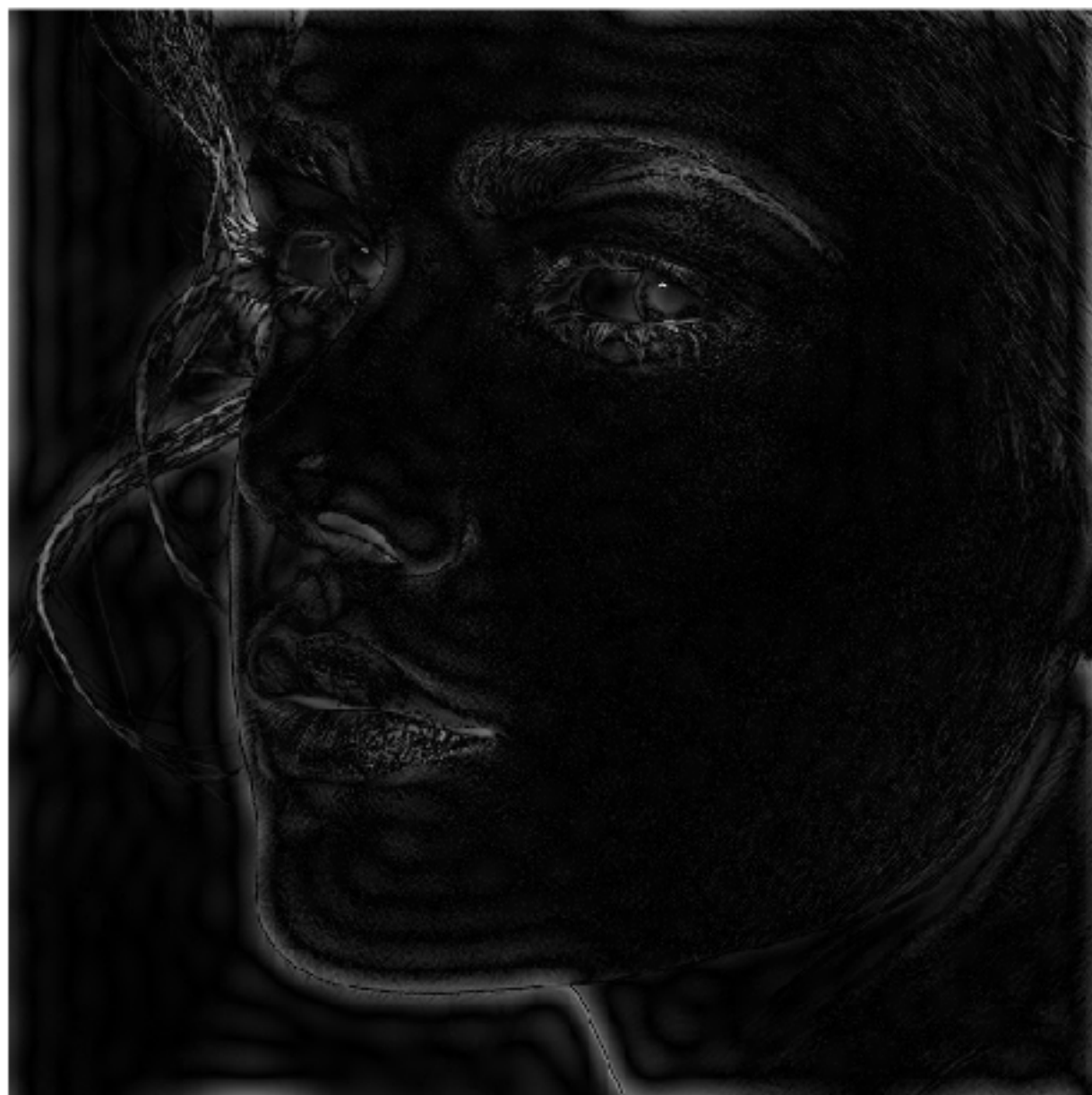


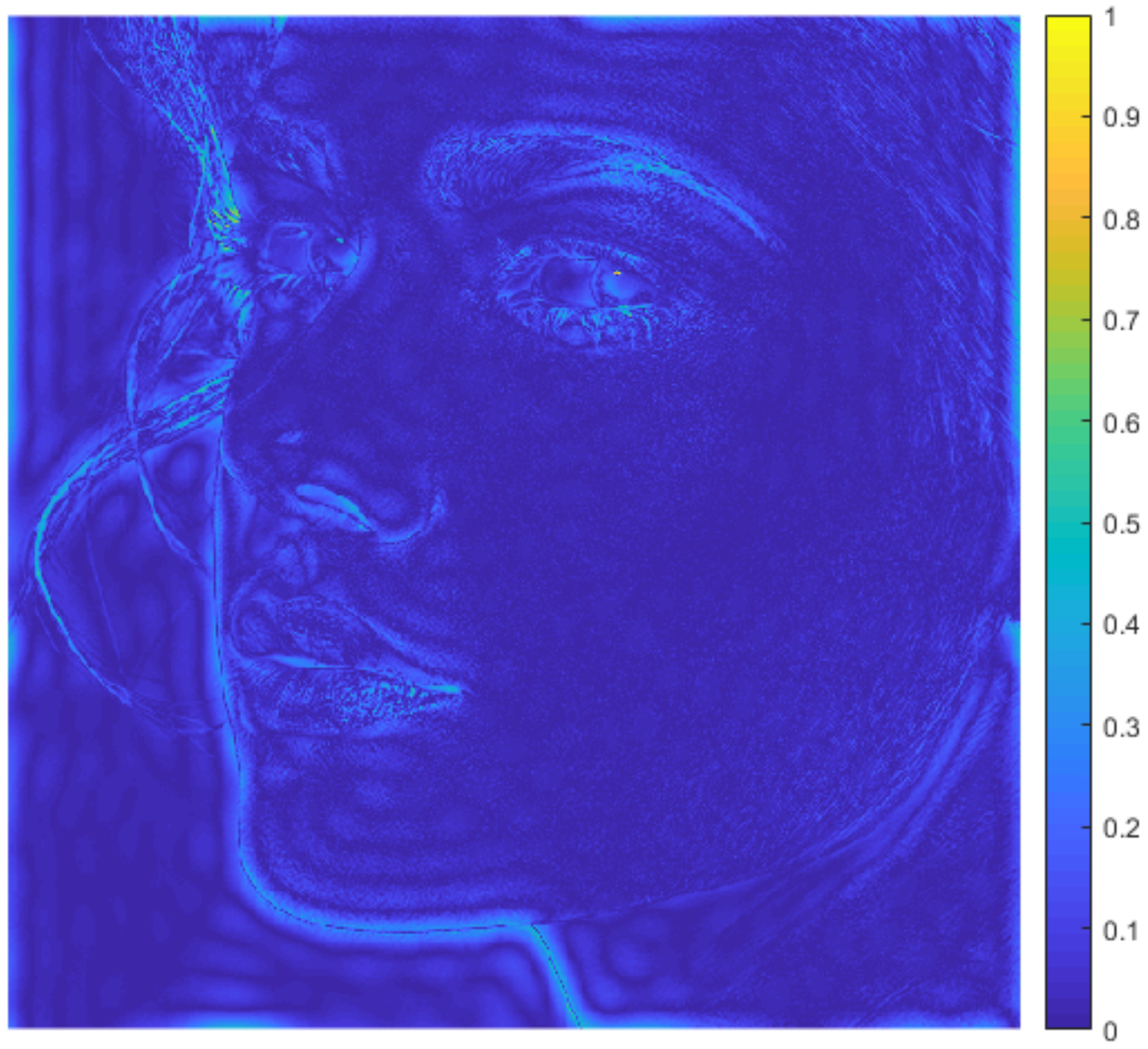




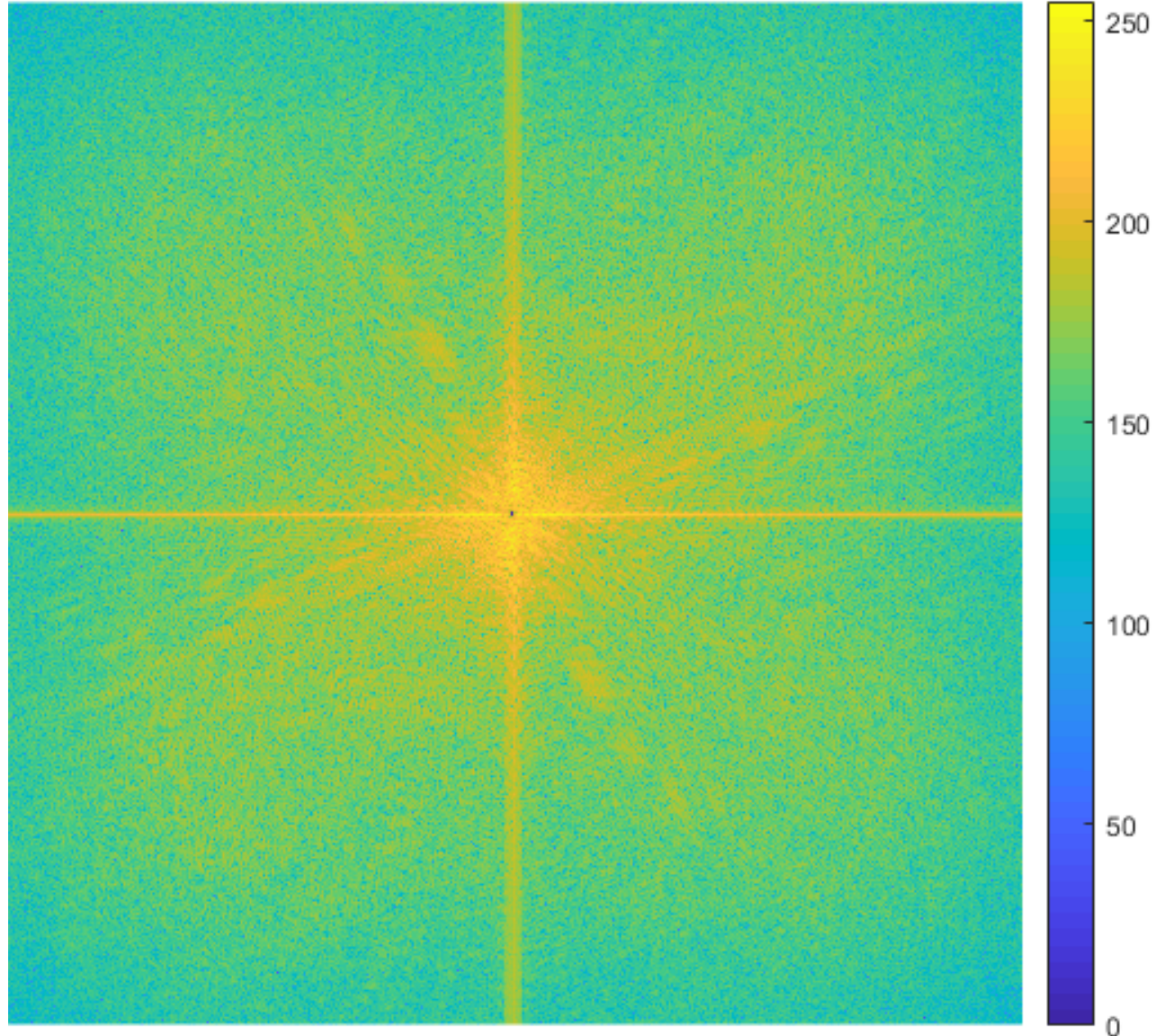


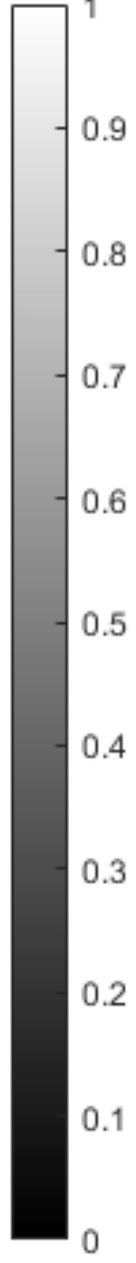


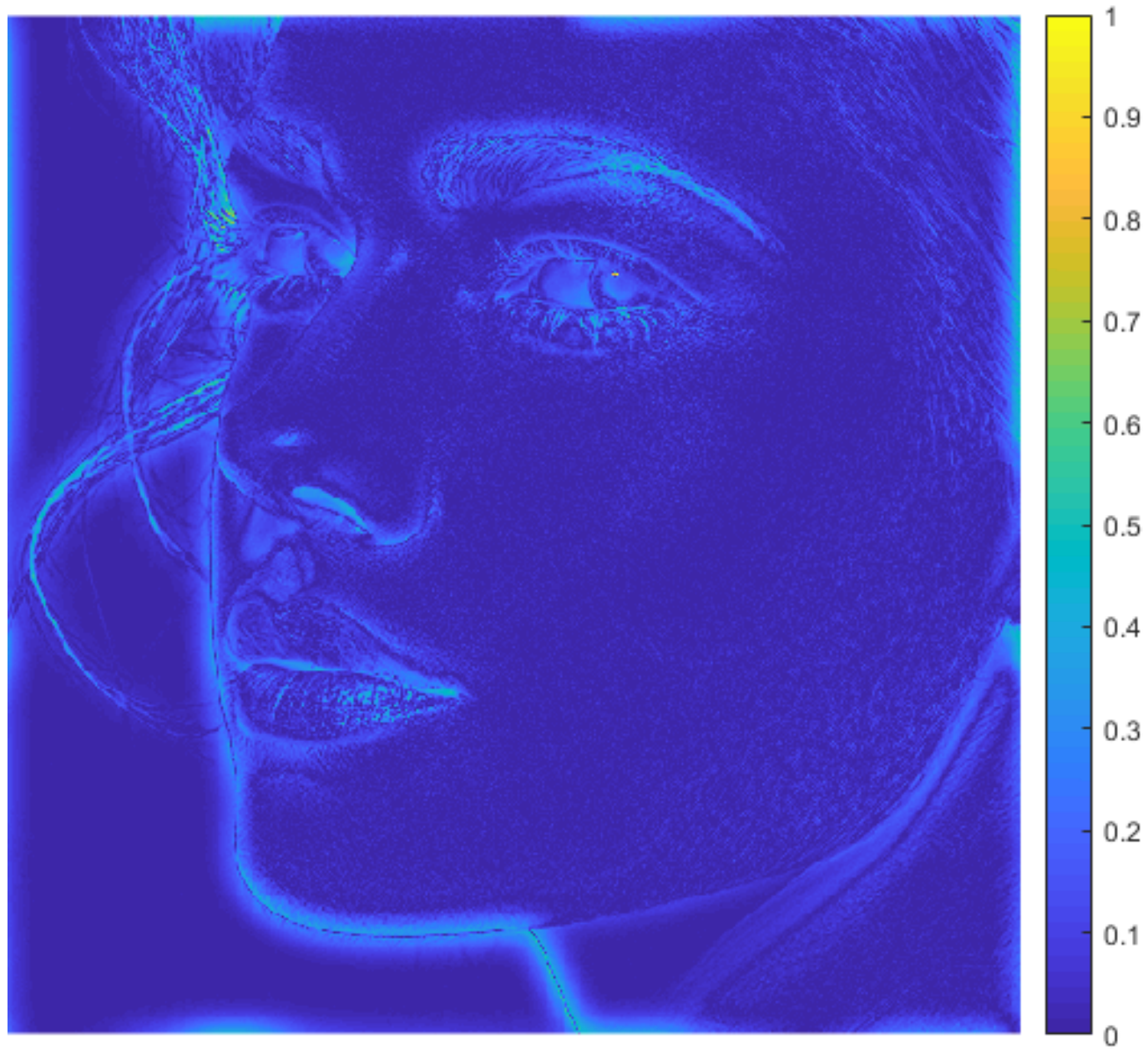


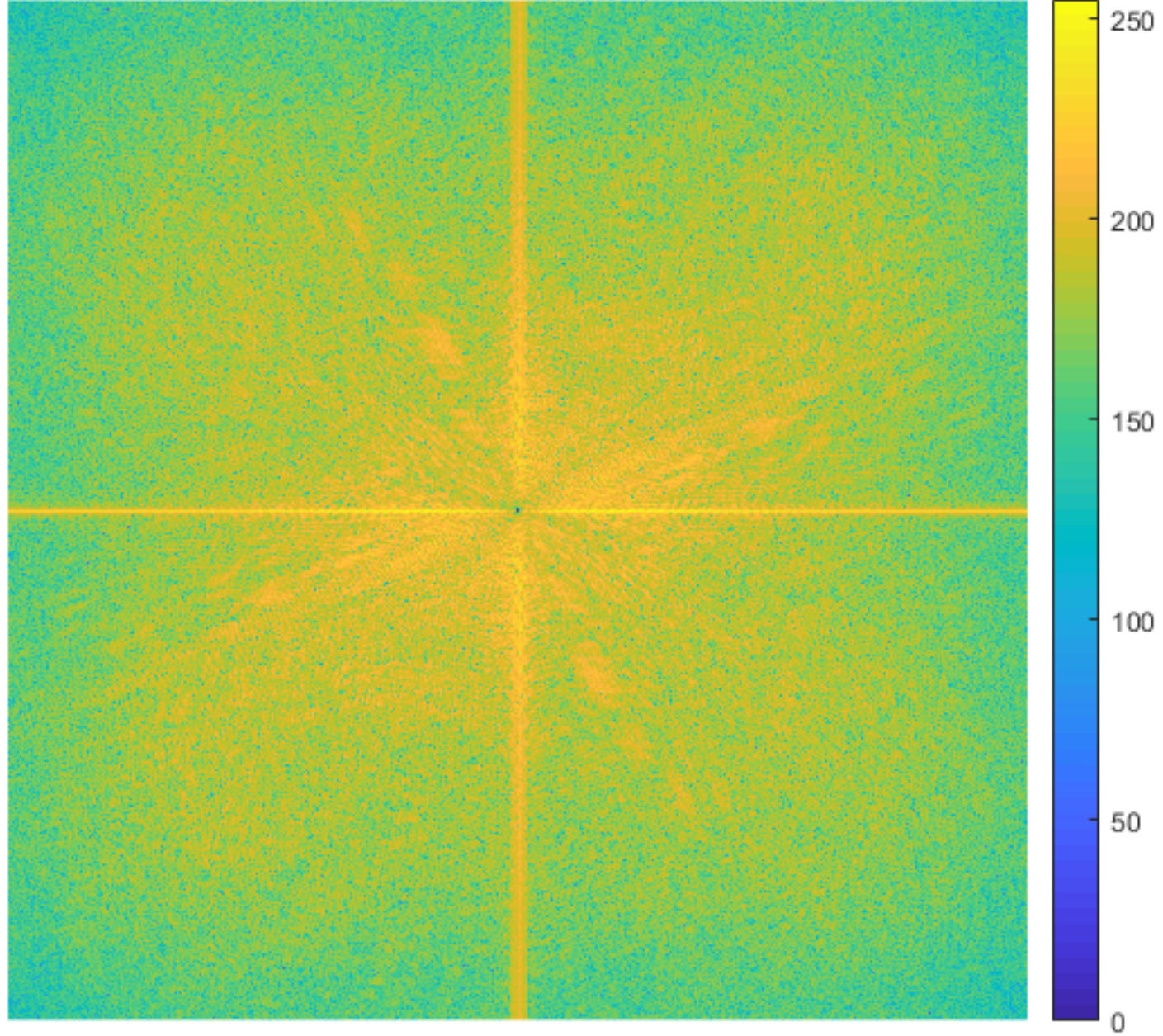


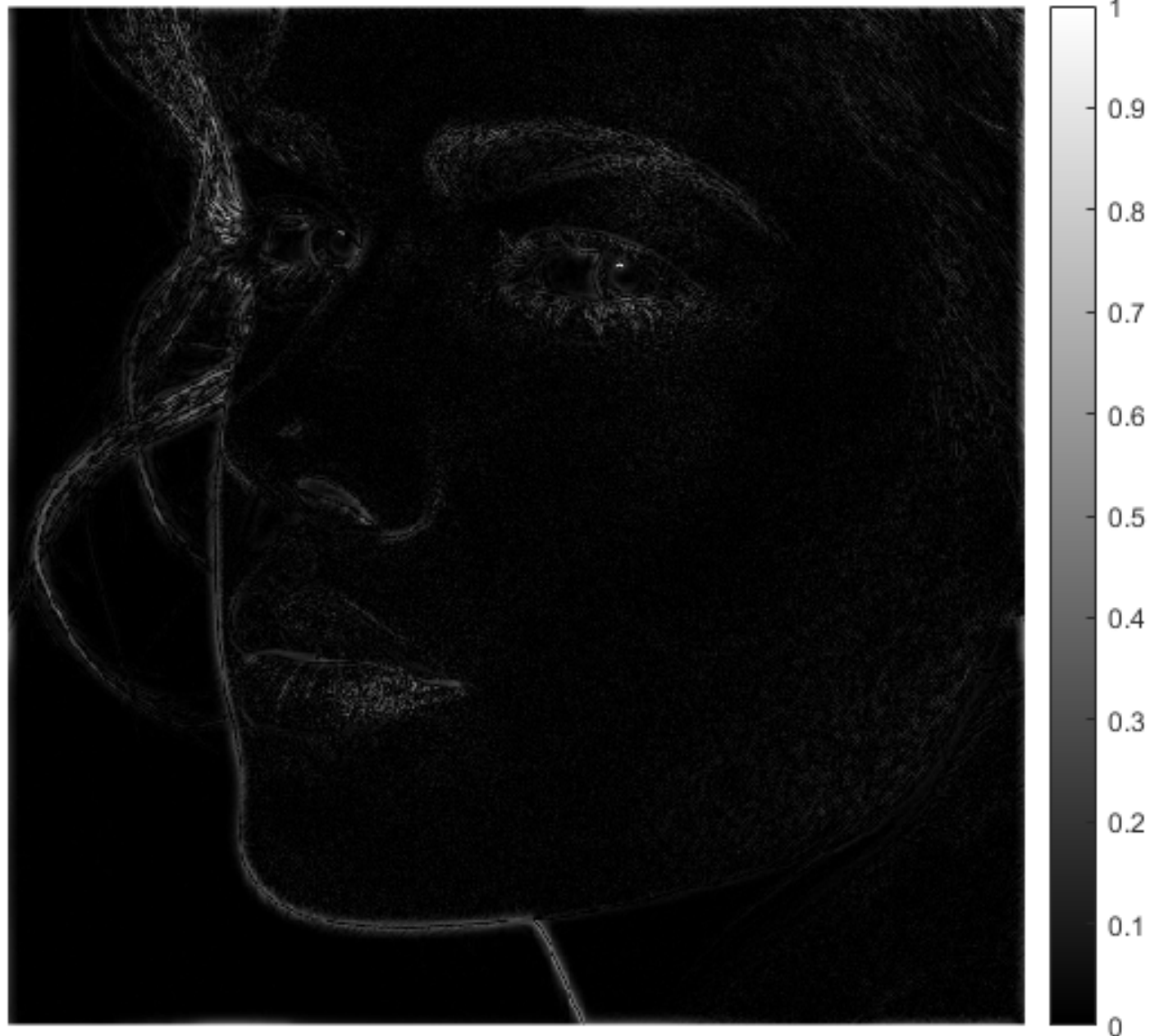


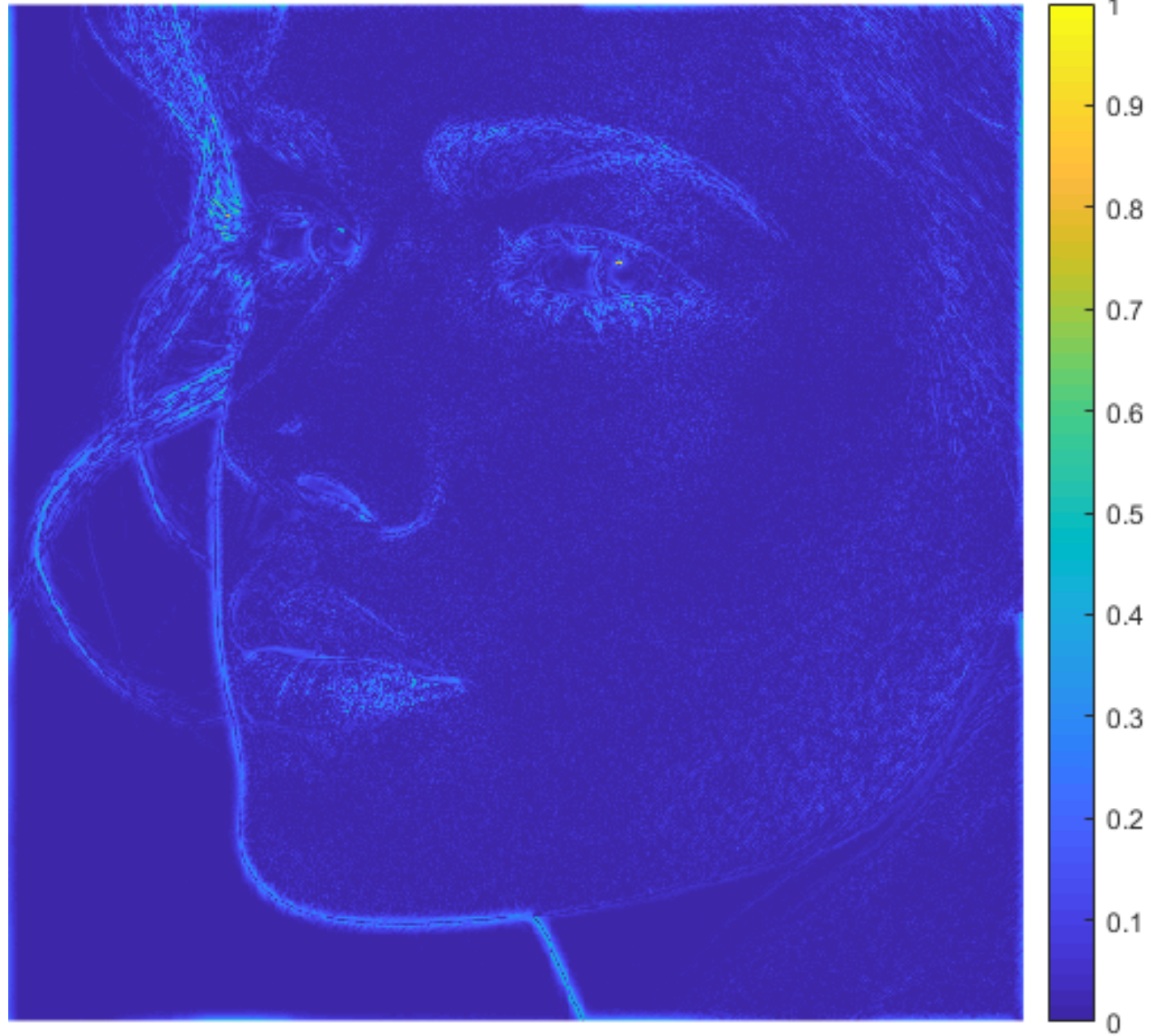








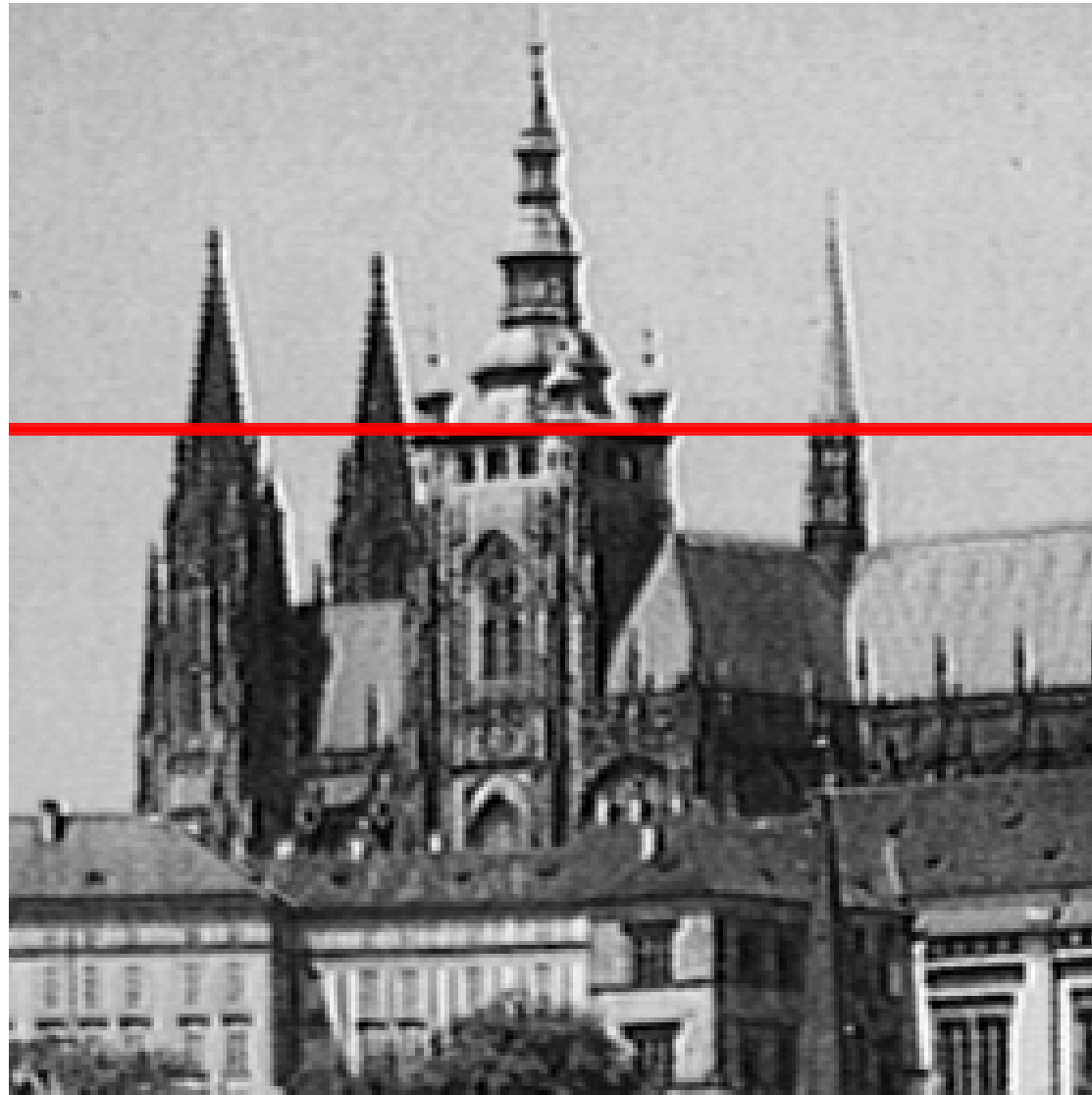




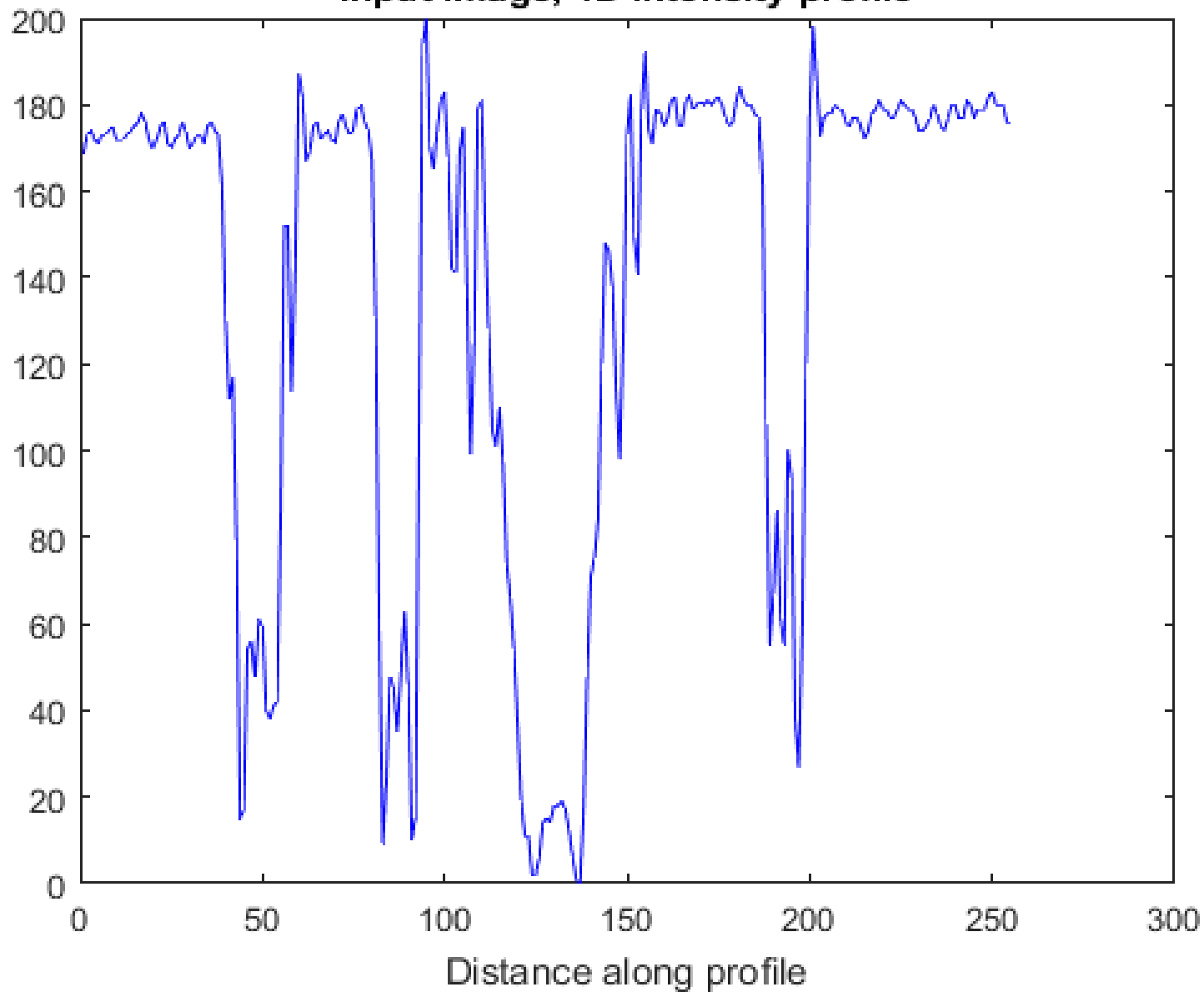
Input image



Input image, line for profile in red



Input image, 1D intensity profile



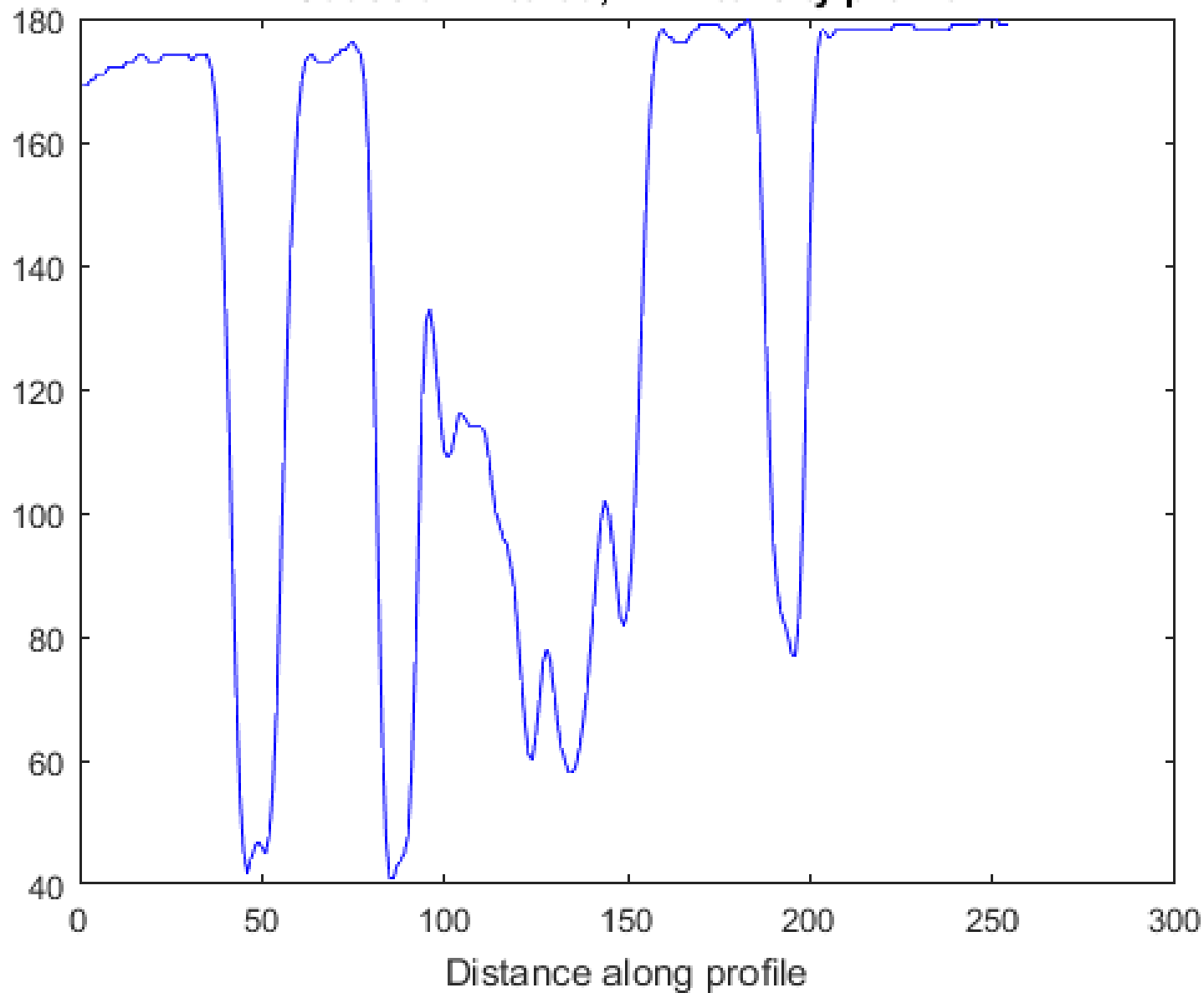
Gaussian filtered



Gaussian filtered, line for profile in red



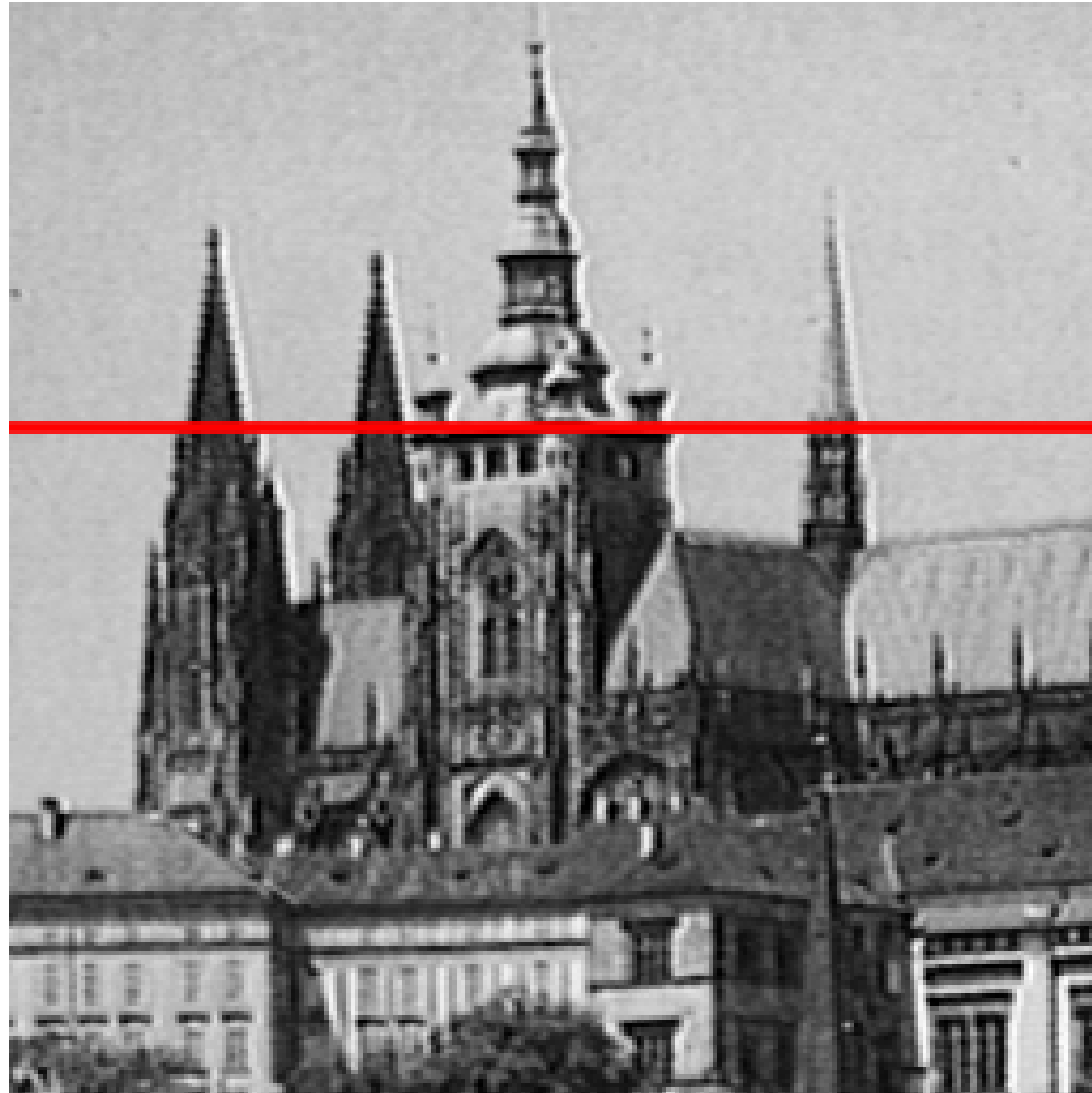
Gaussian filtered, 1D intensity profile



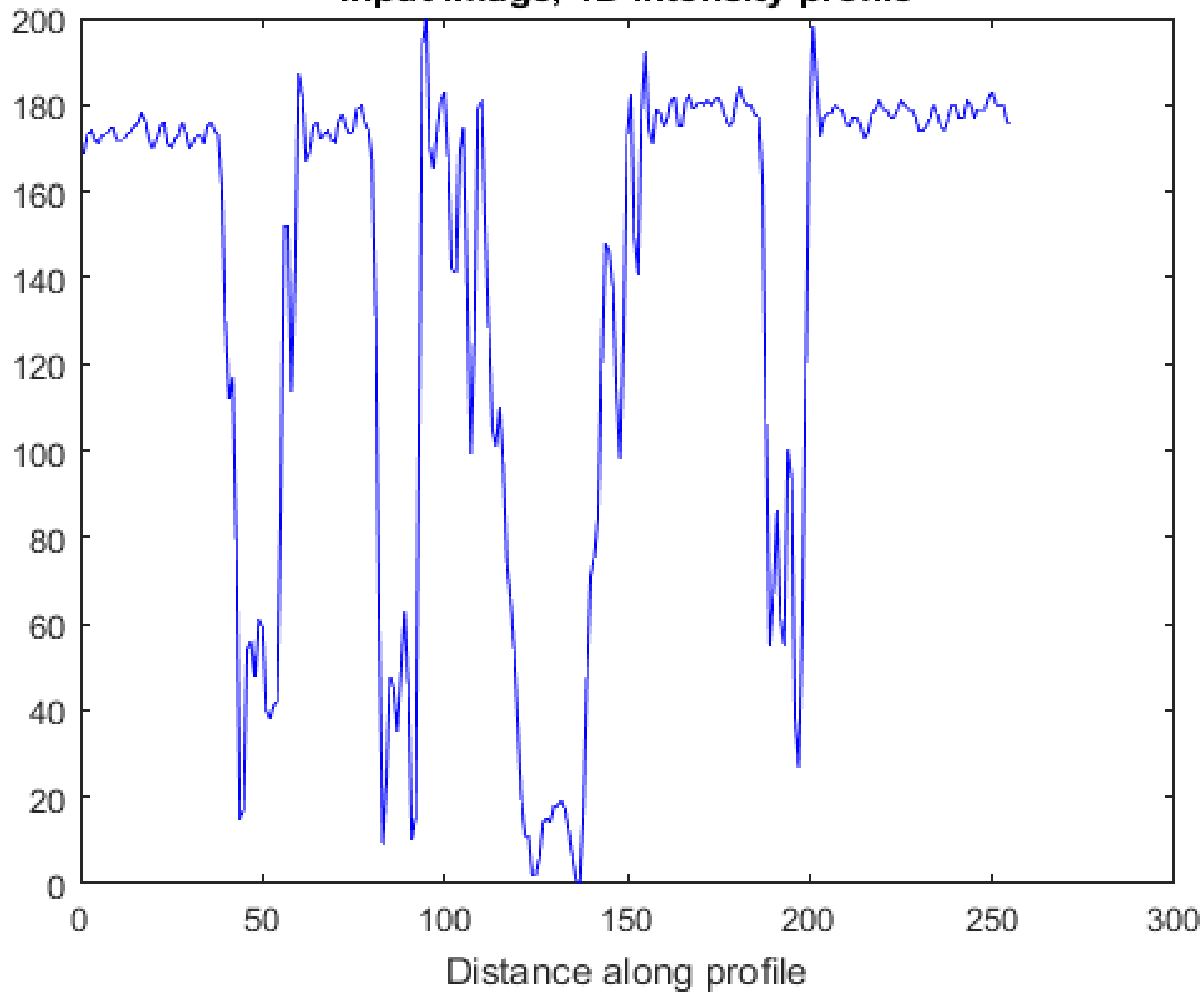
Input image



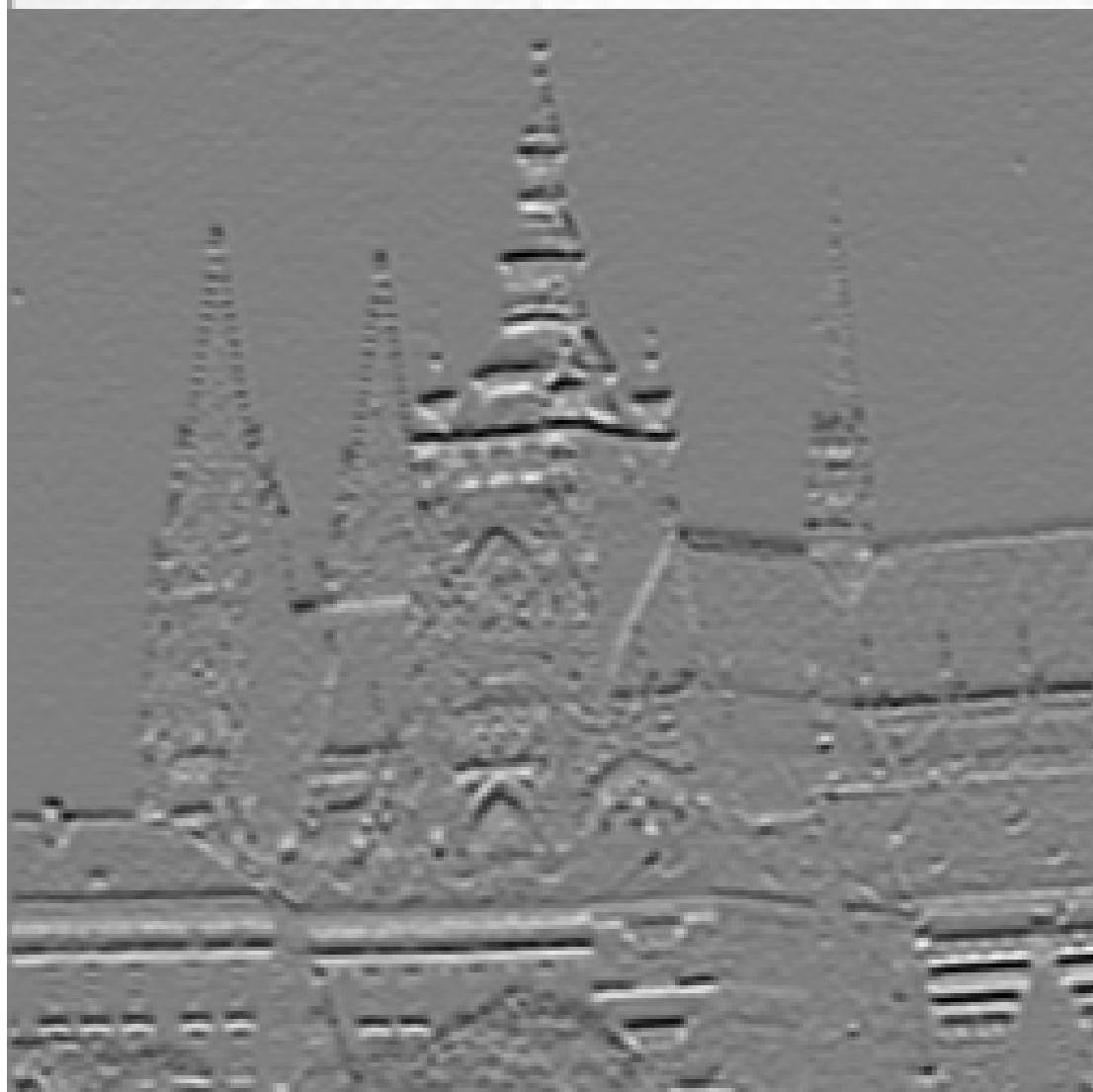
Input image, line for profile in red



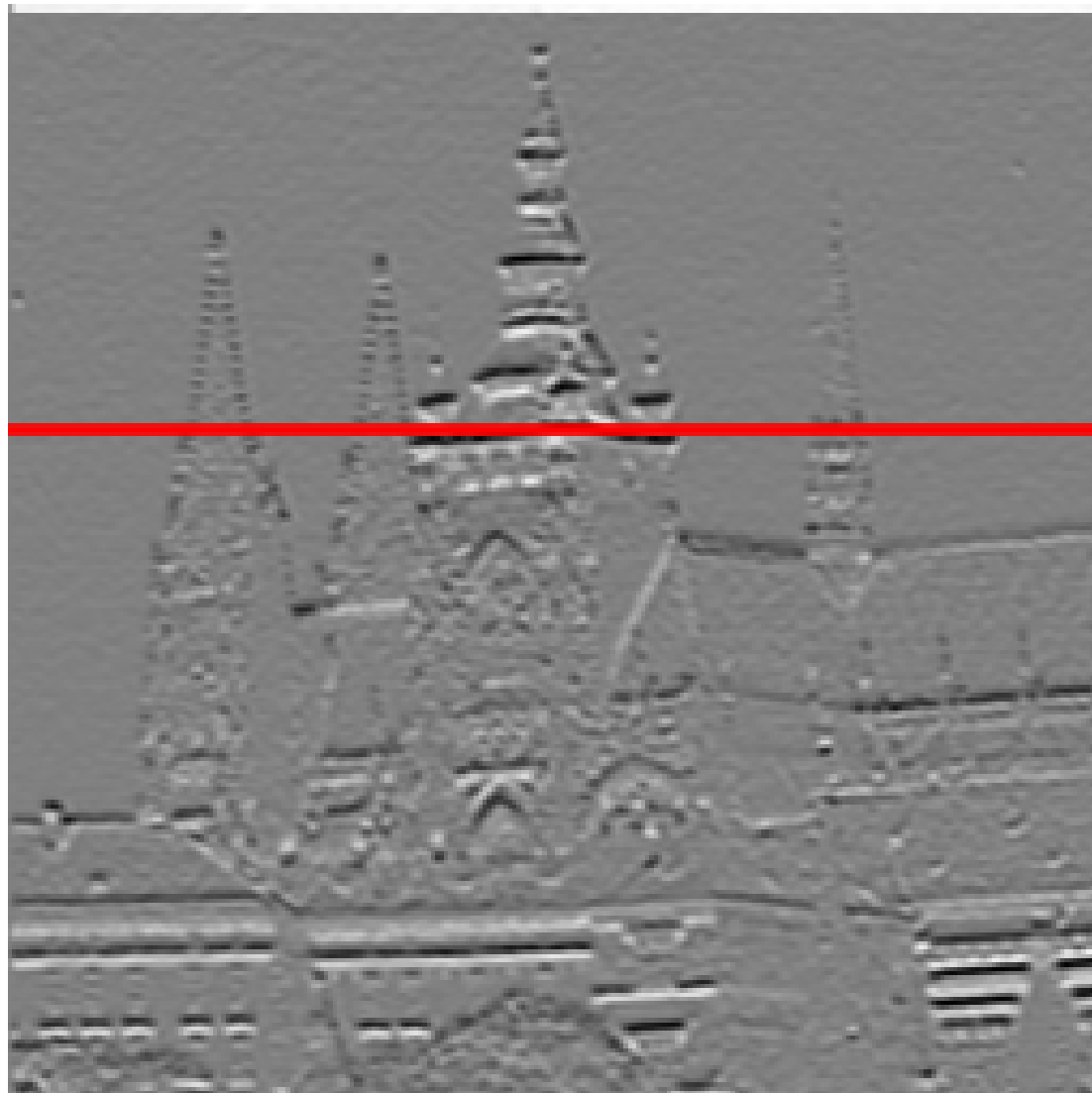
Input image, 1D intensity profile



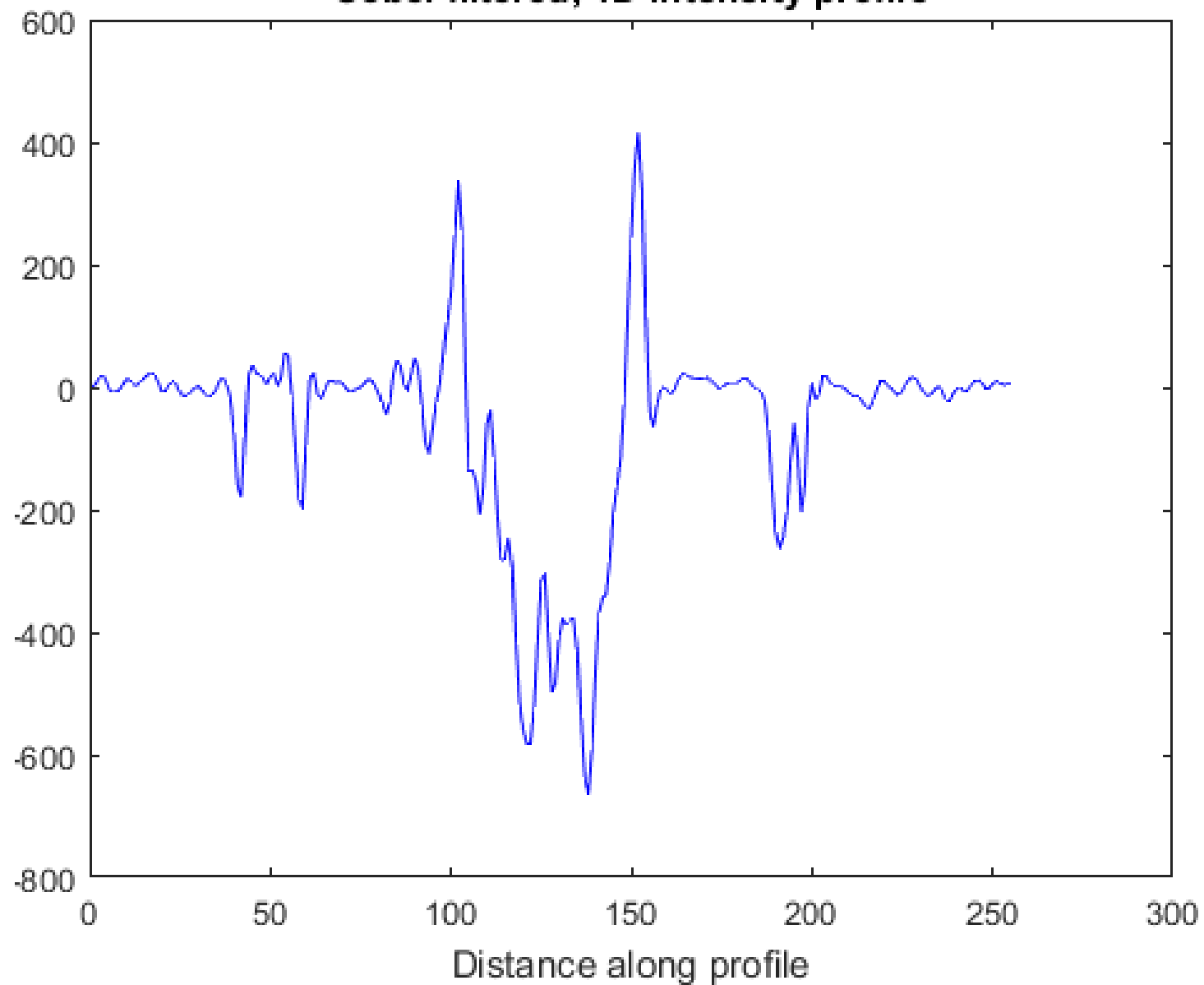
Sobel filtered



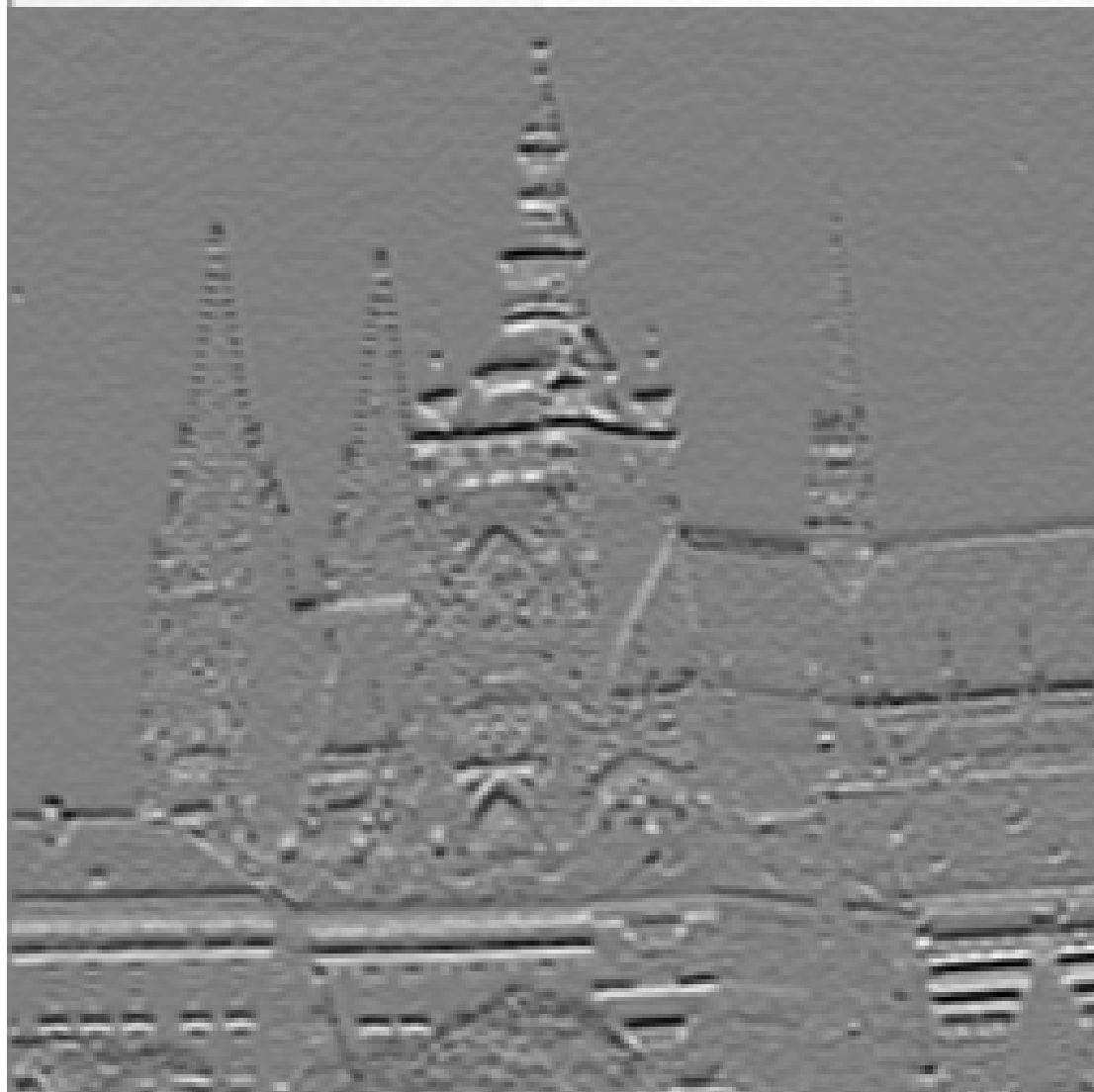
Sobel filtered, line for profile in red



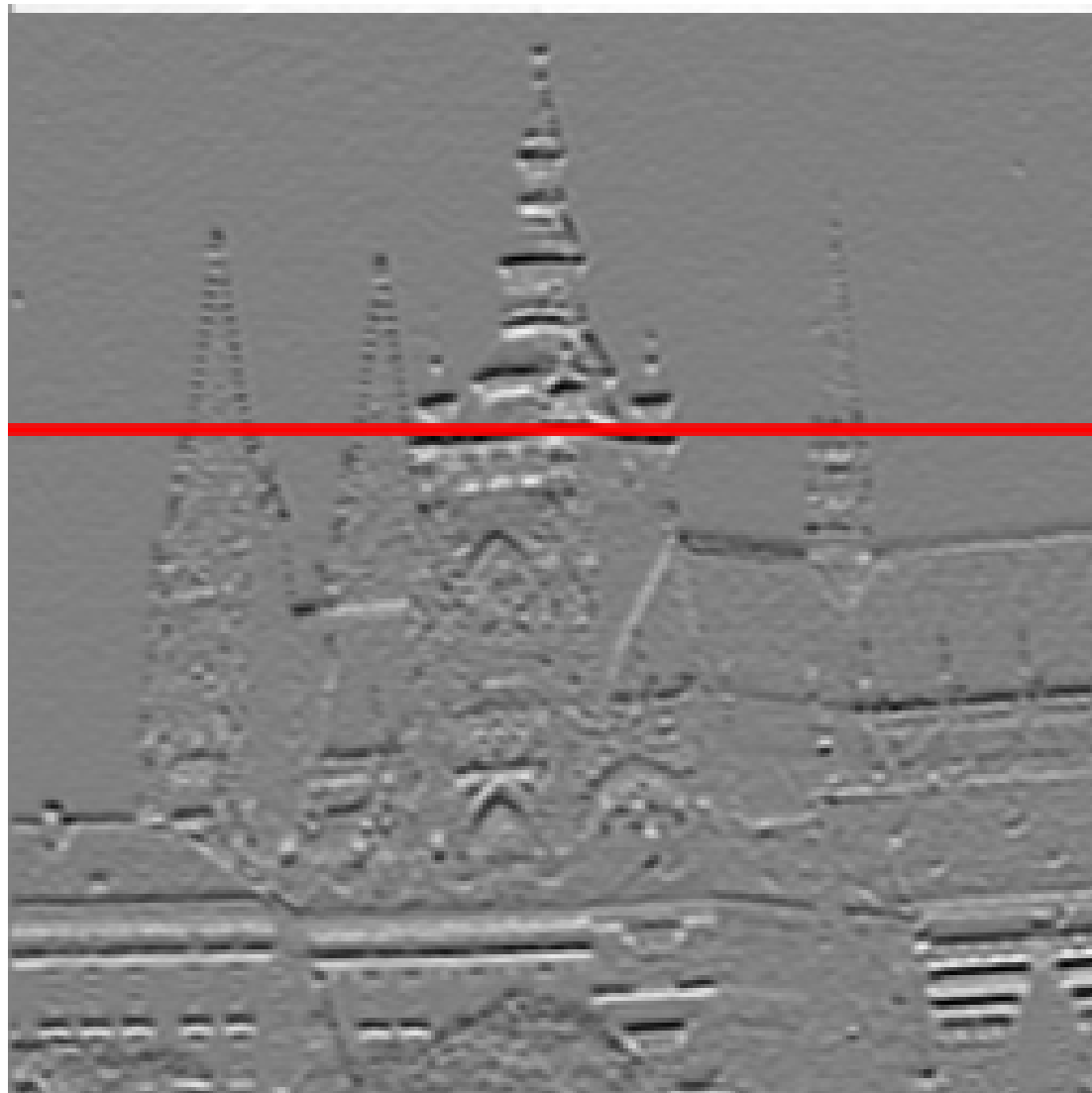
Sobel filtered, 1D intensity profile



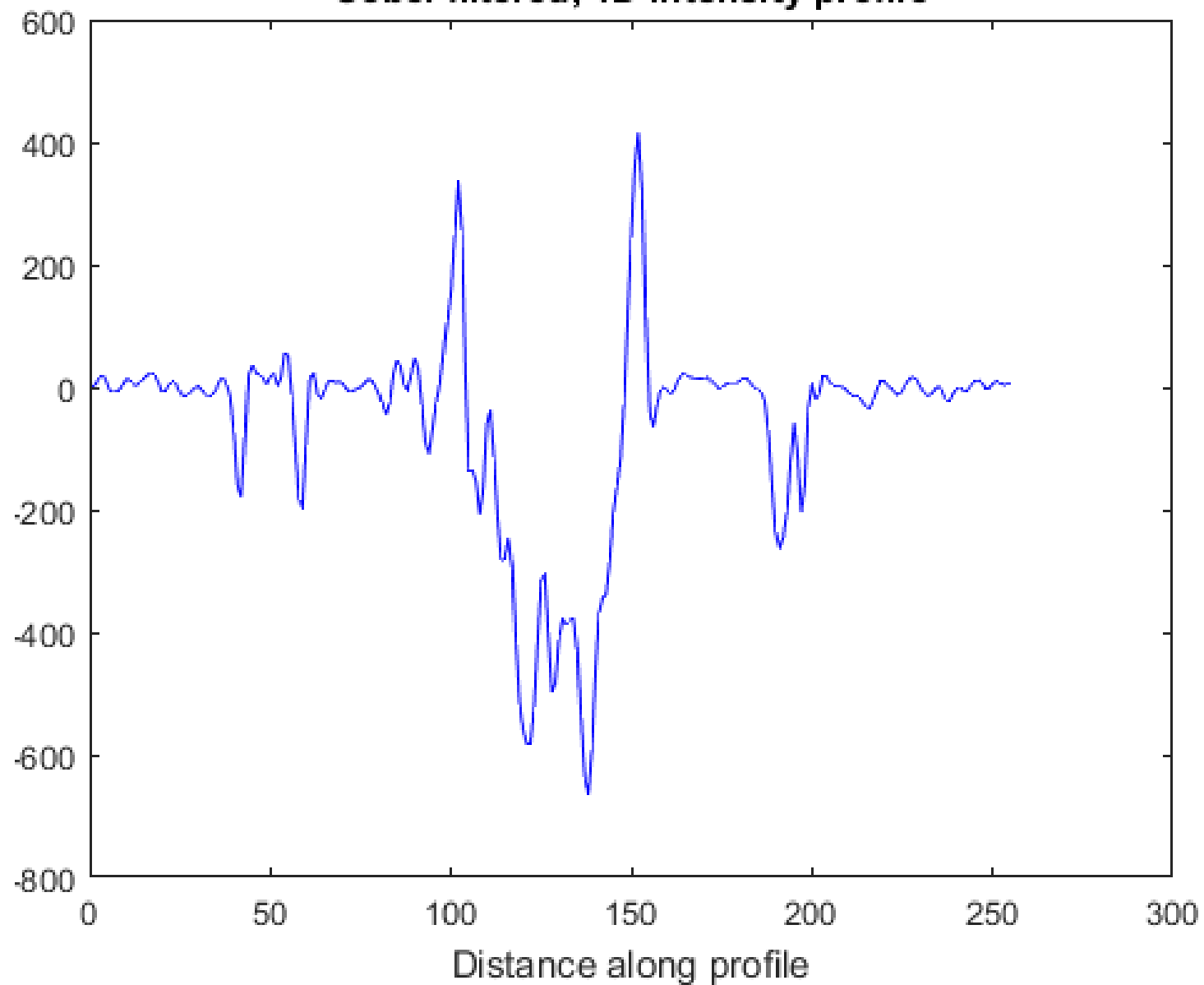
Sobel filtered



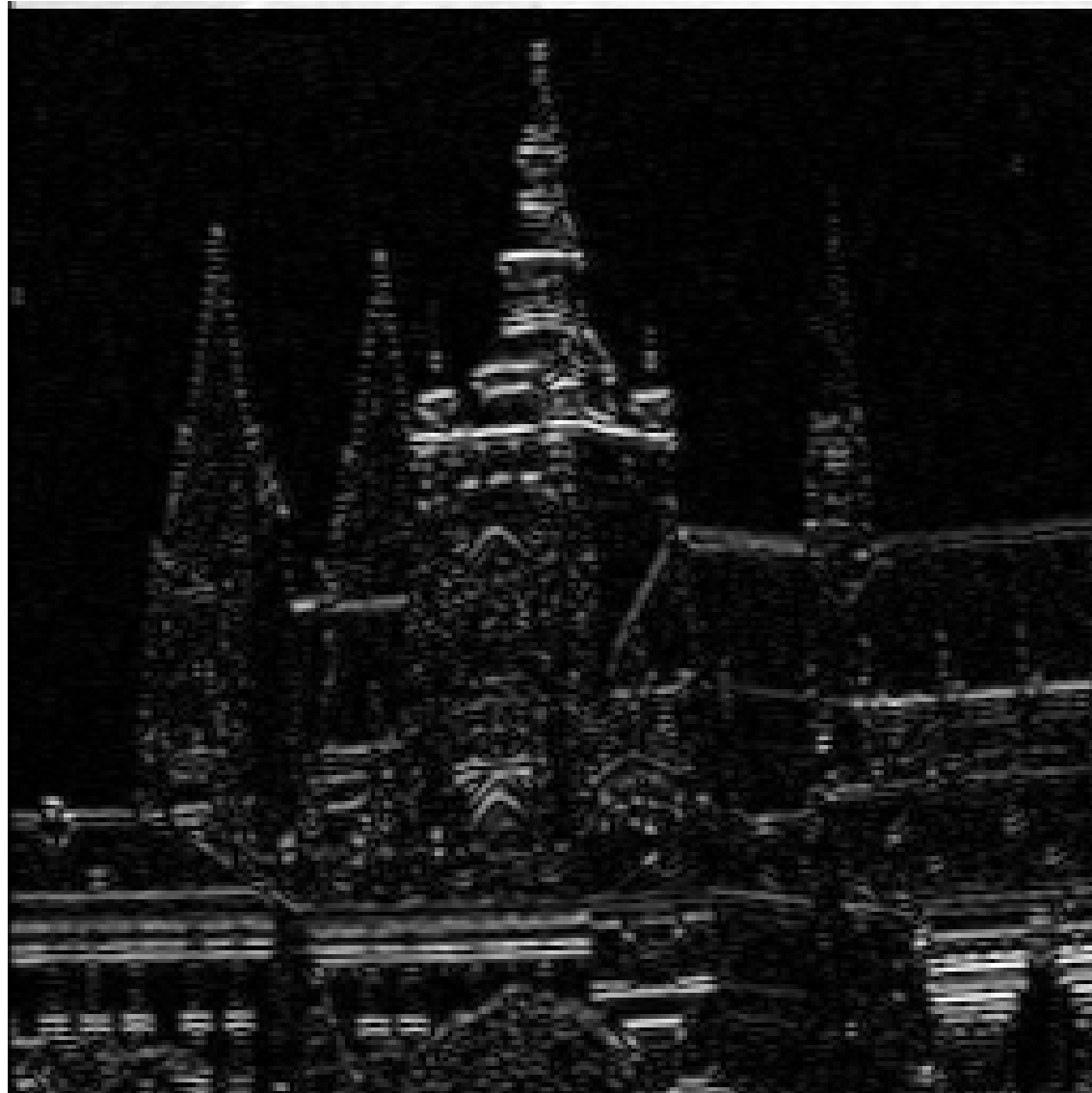
Sobel filtered, line for profile in red



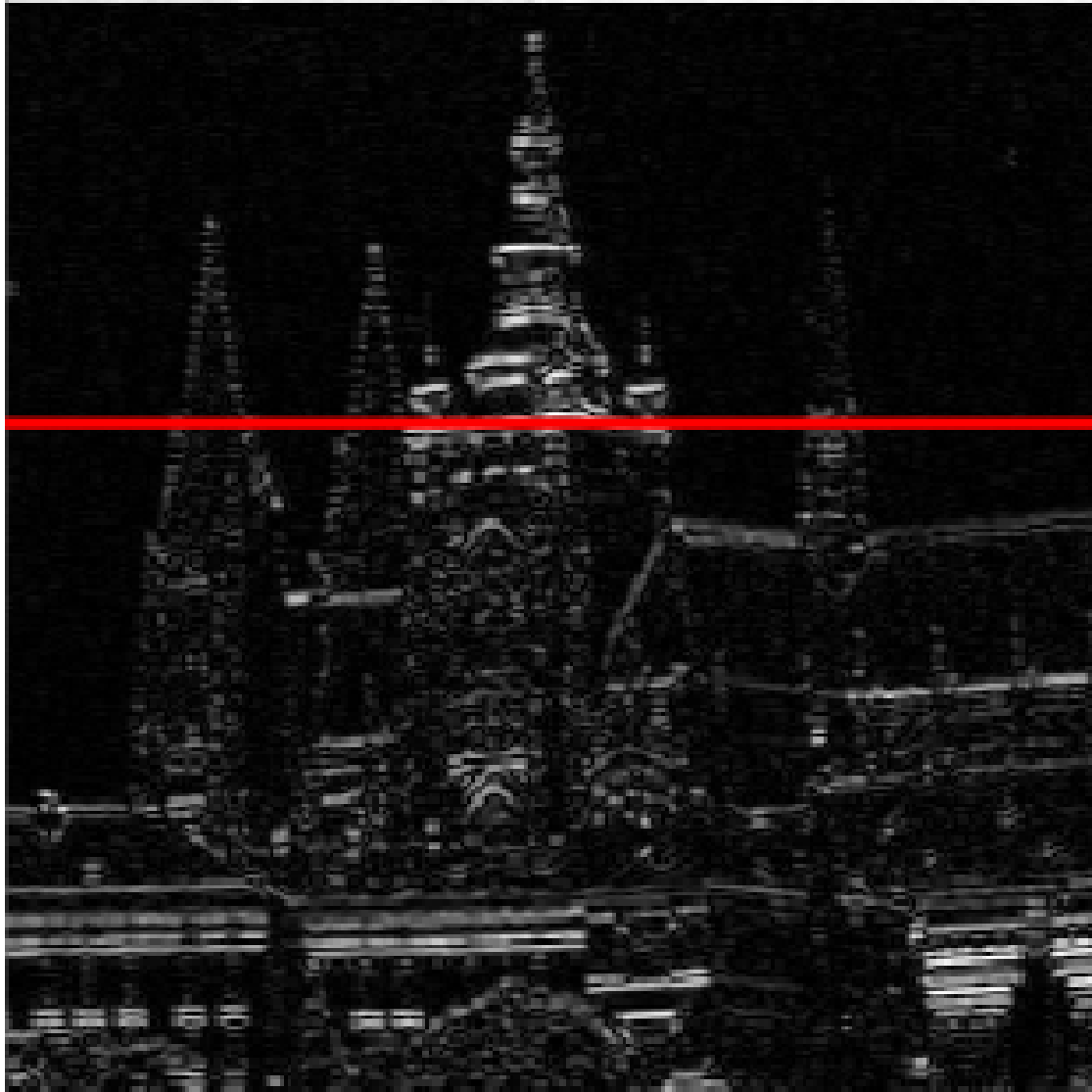
Sobel filtered, 1D intensity profile



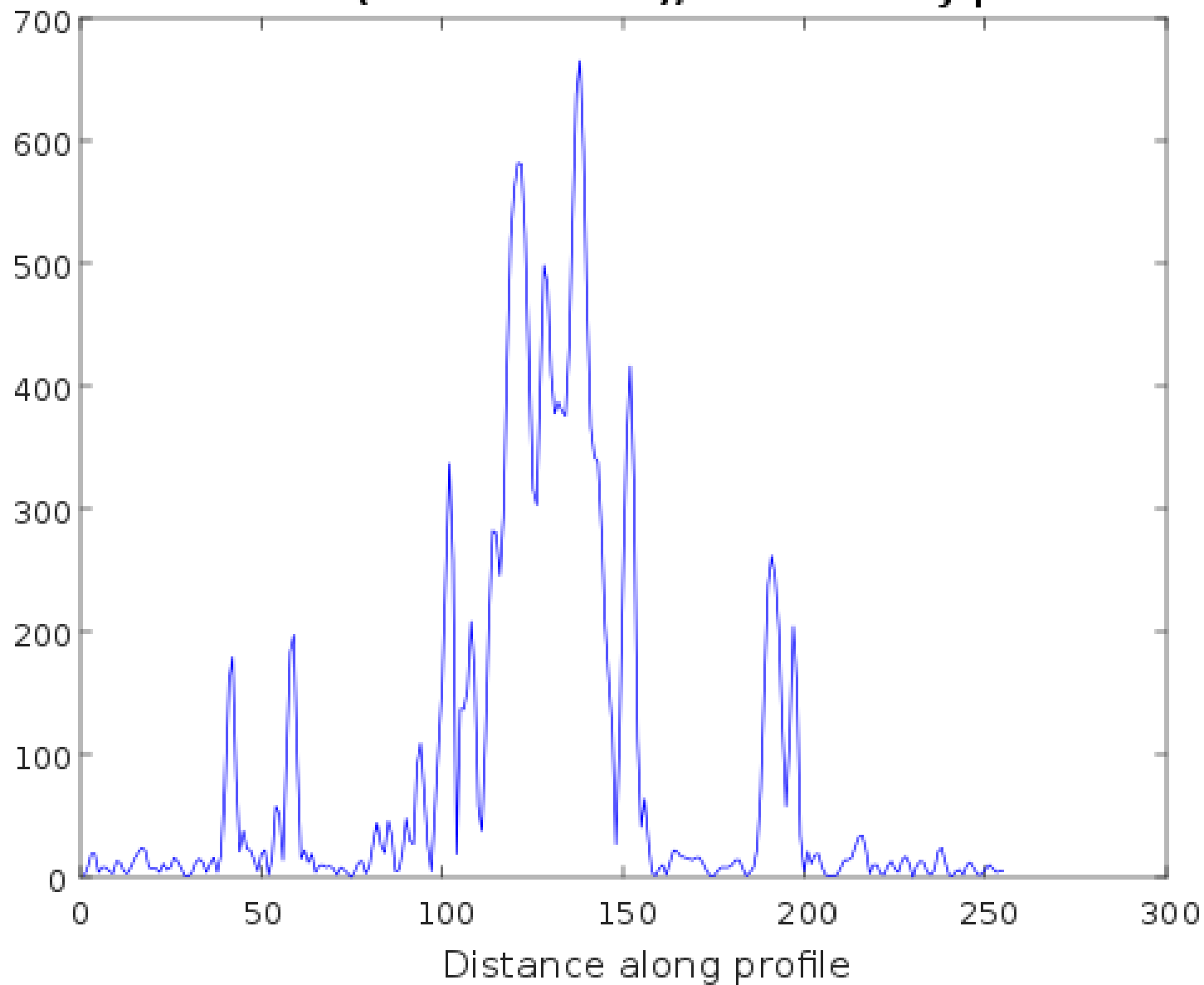
Abs(Sobel filtered)



Abs(Sobel filtered)

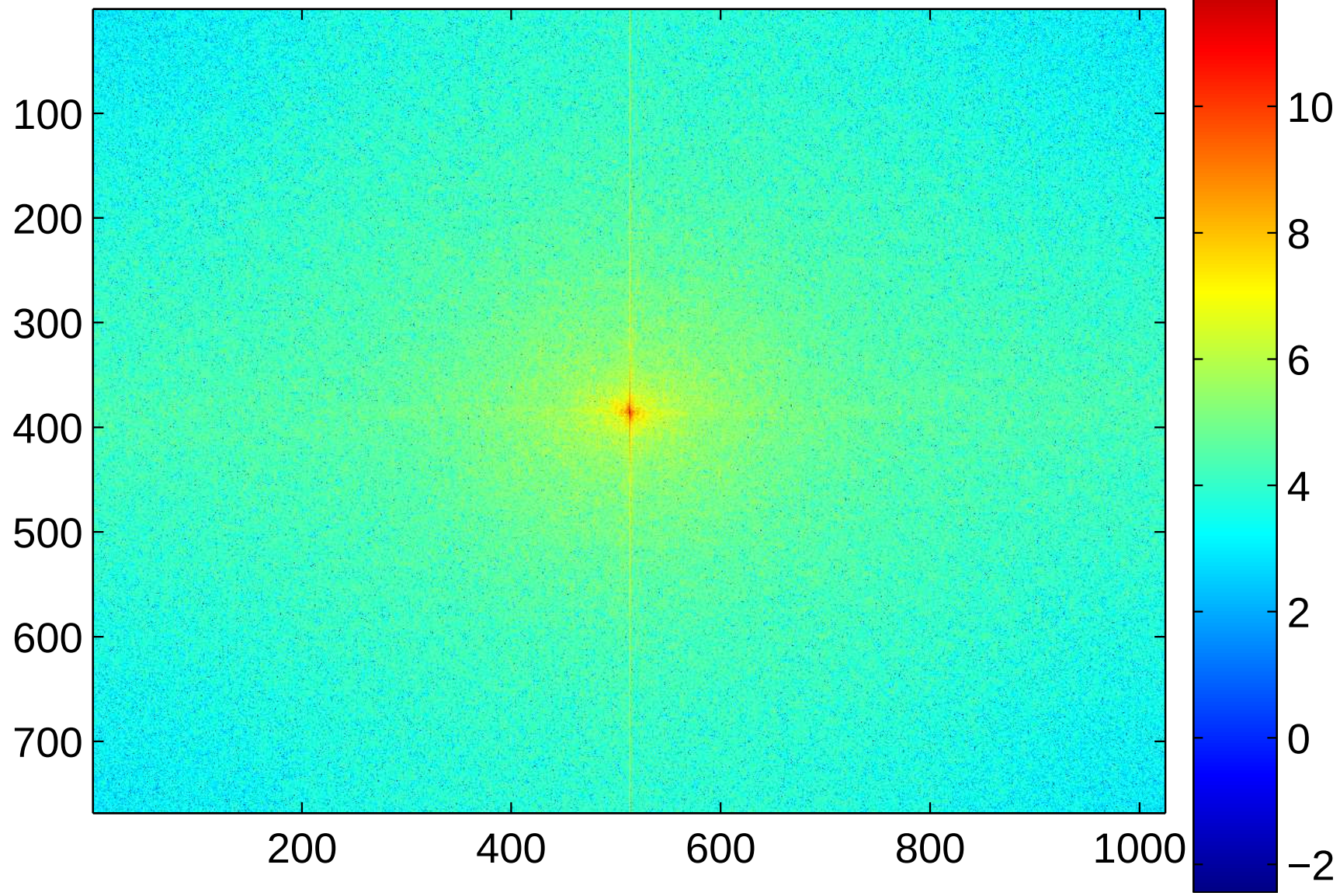


Abs(Sobel filtered), 1D intensity profile

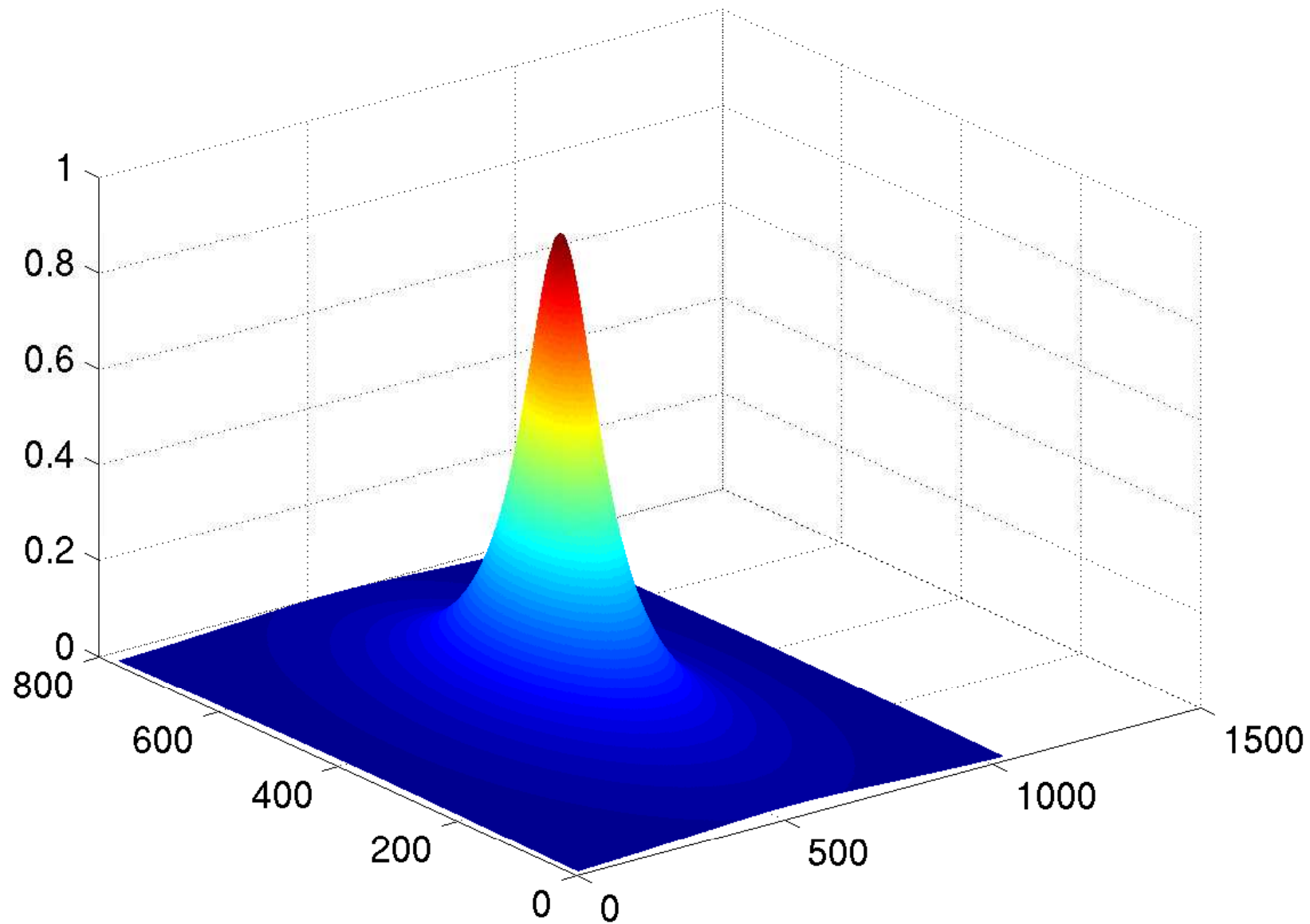




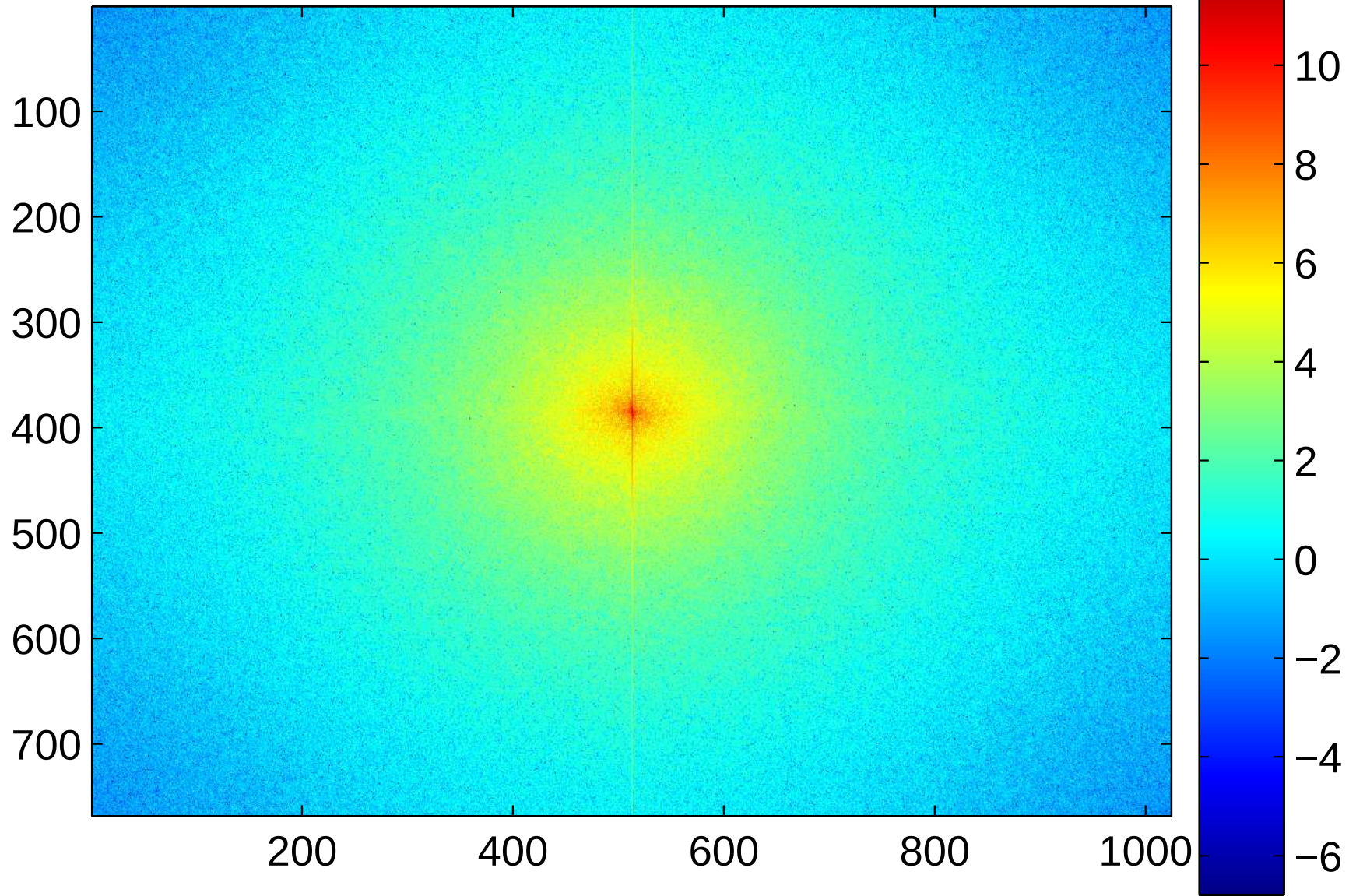
Shifted $\log(\text{abs}(\text{FFT}))$ of the original image



lp Butt filter n=1, cutoff=66



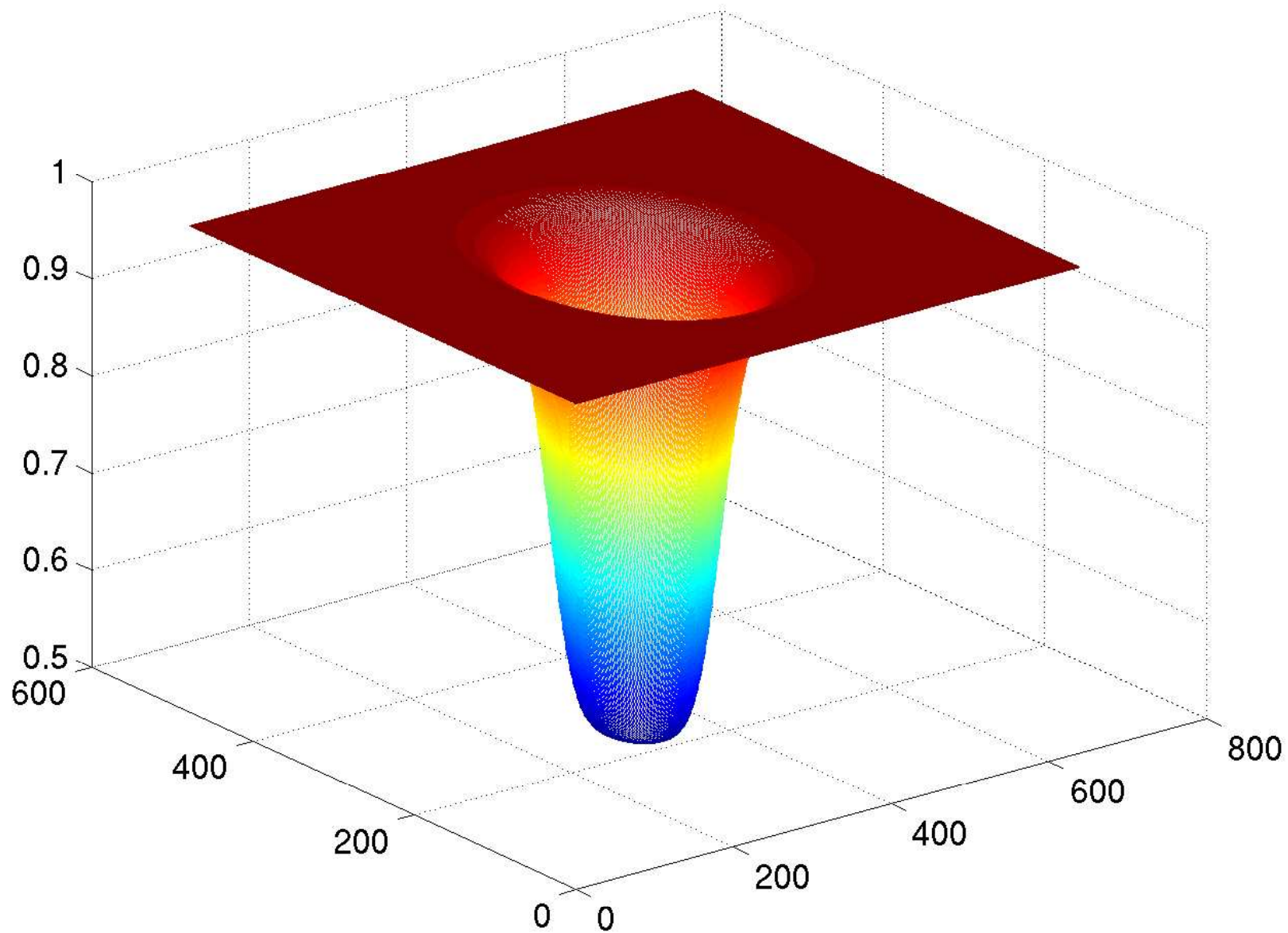
Shifted $\log(\text{abs}(\text{FFT}))$ of the filtered image

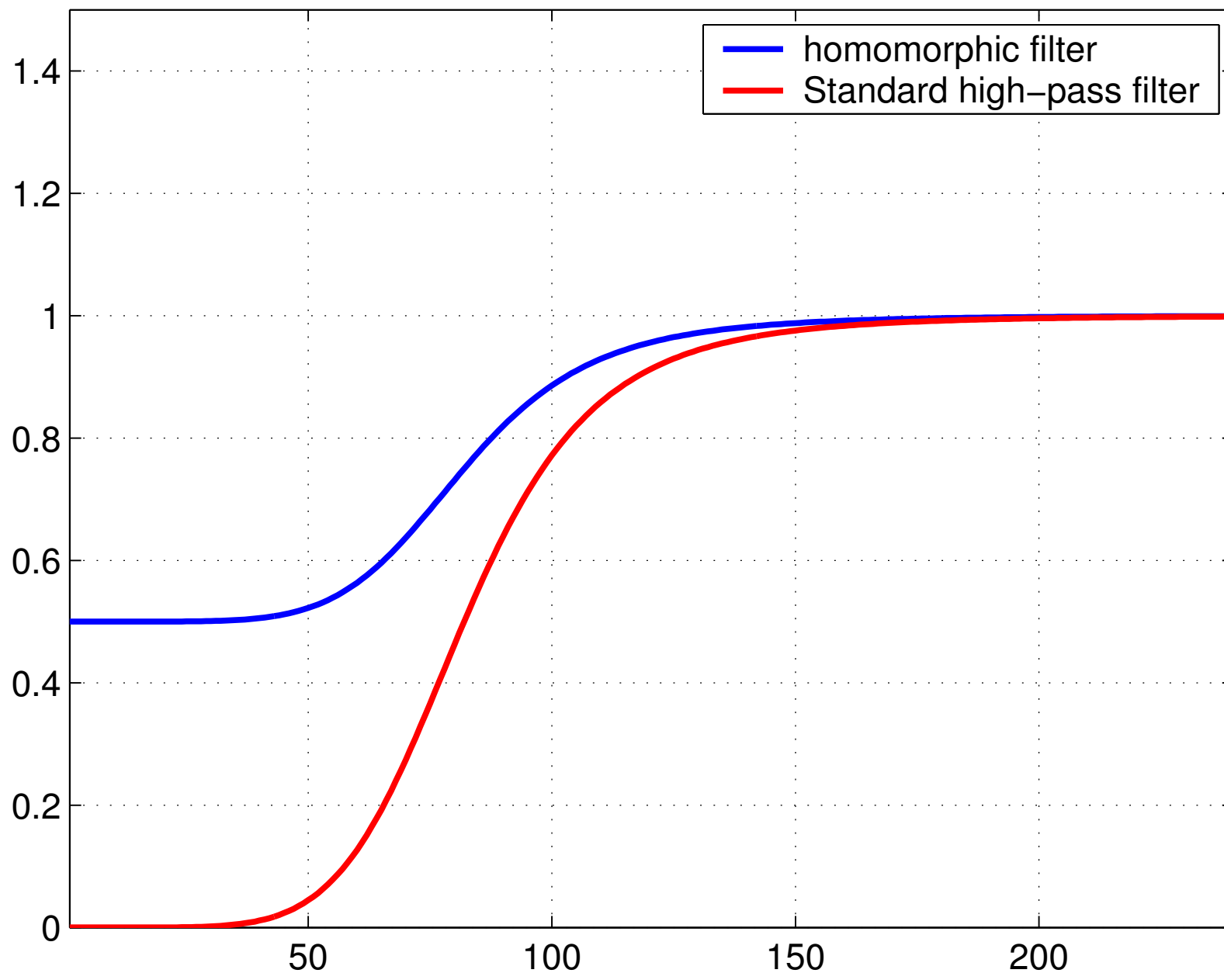






Homomorphic filter made by adaptation of Buttherworth highpass

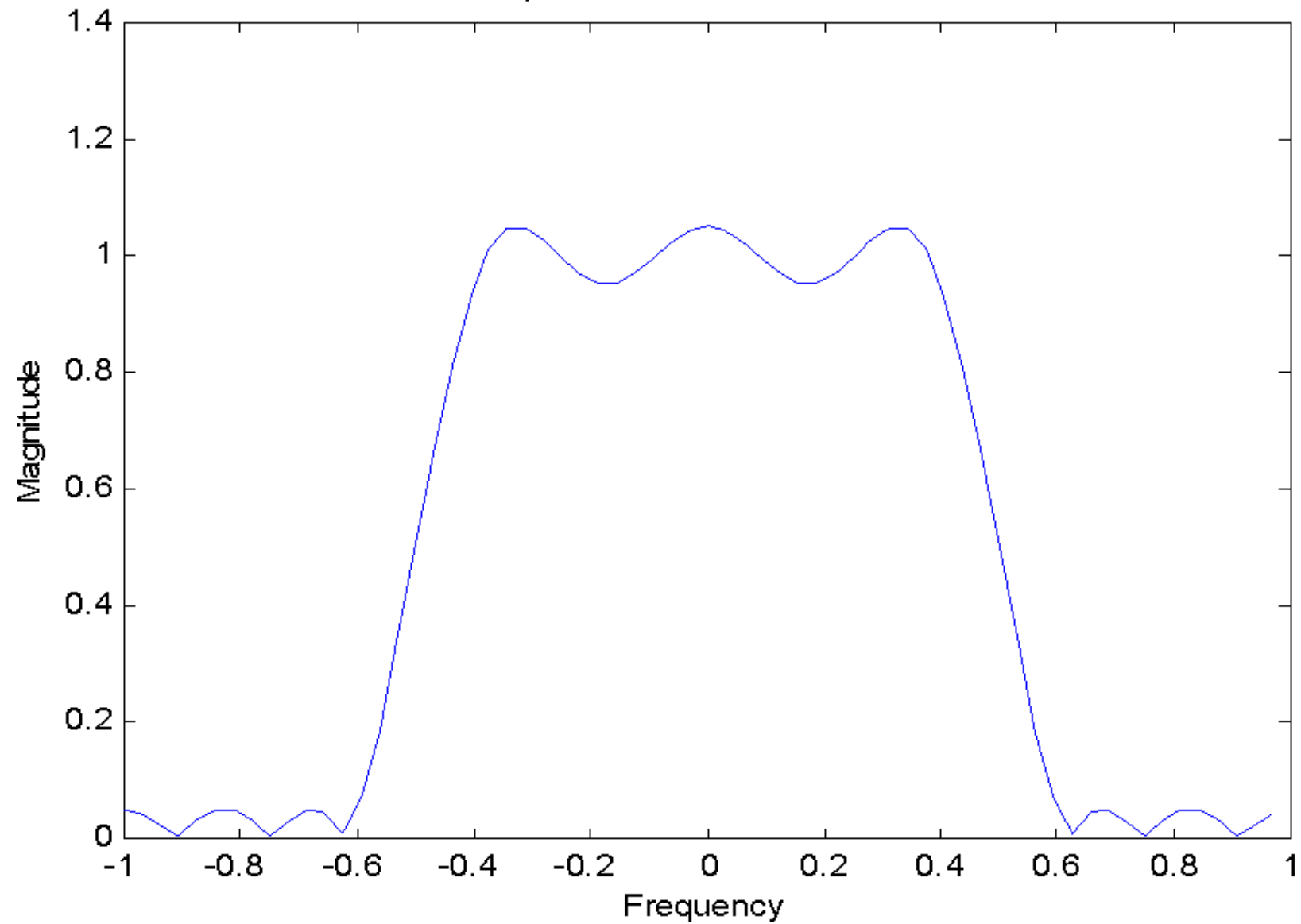




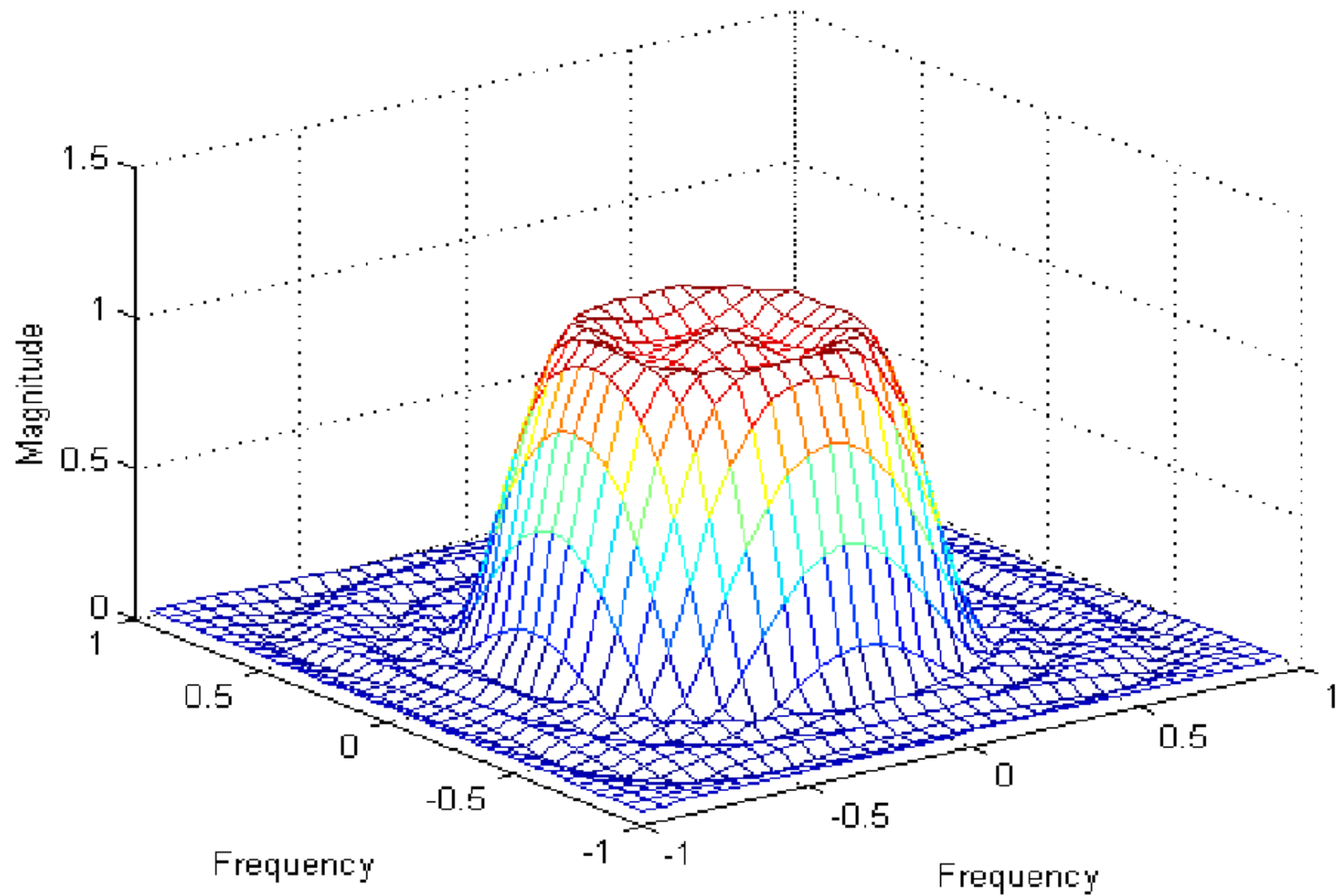




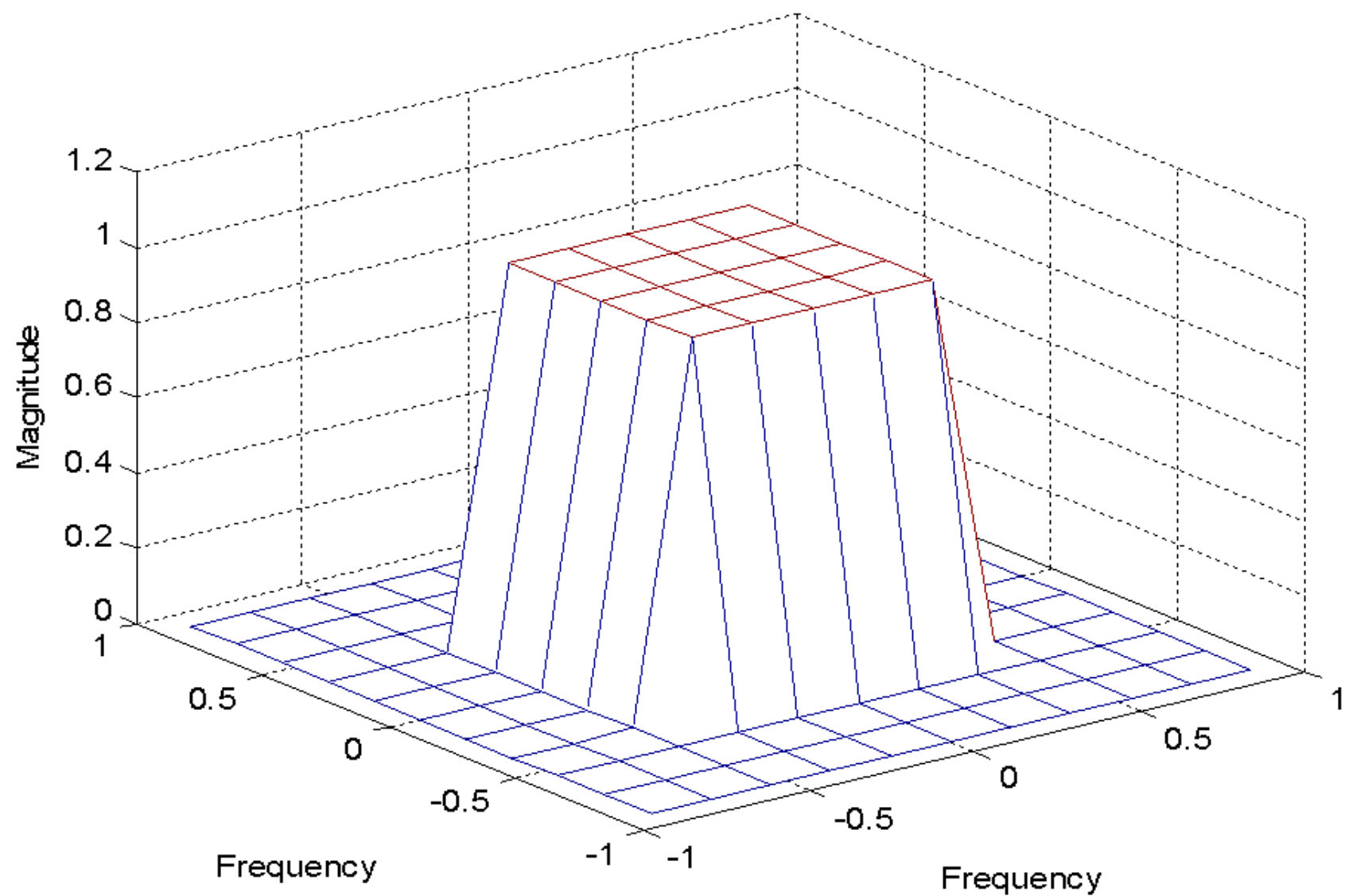
1D optimal Parks-McClellan filter



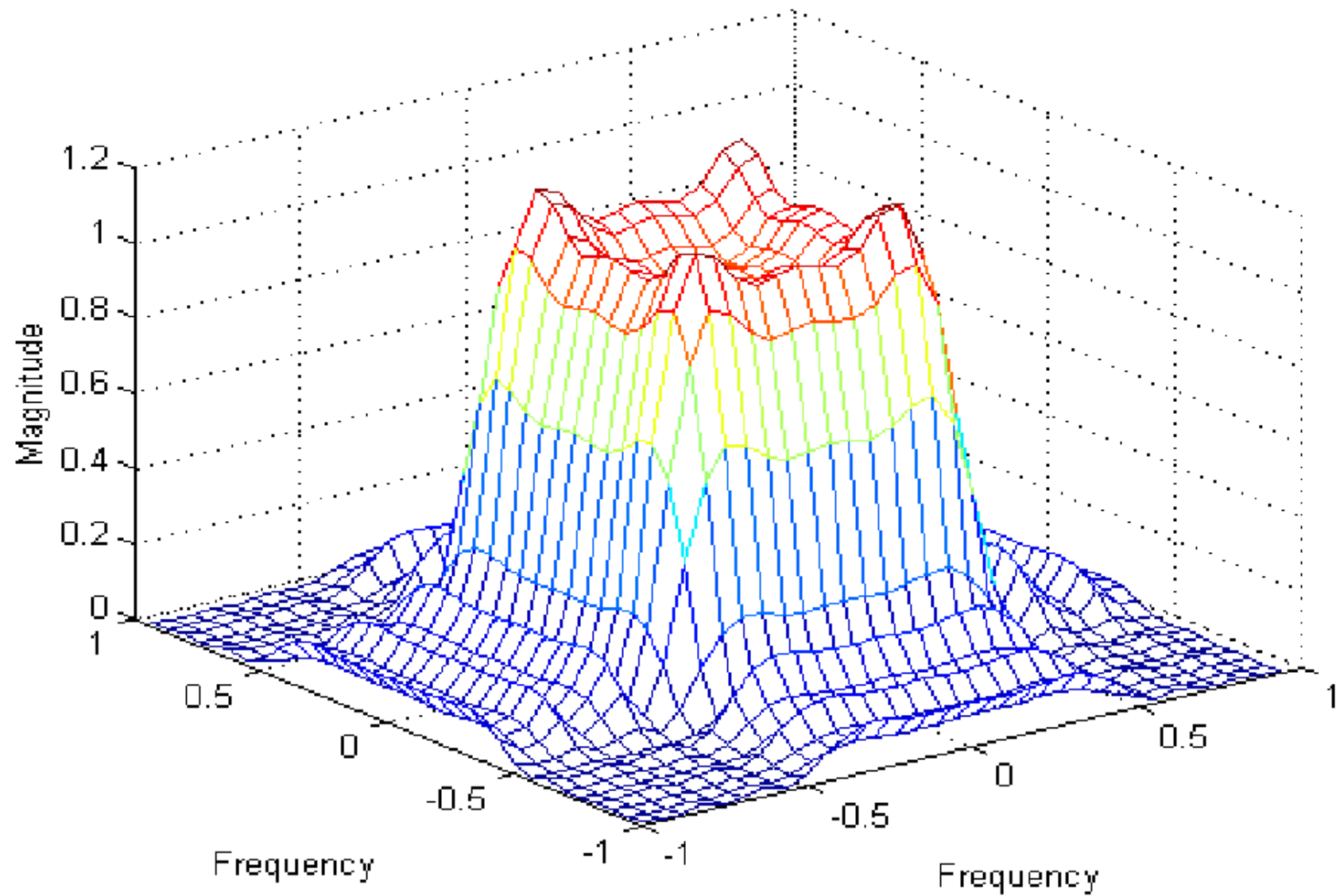
1D optimal Parks-McClellan filter



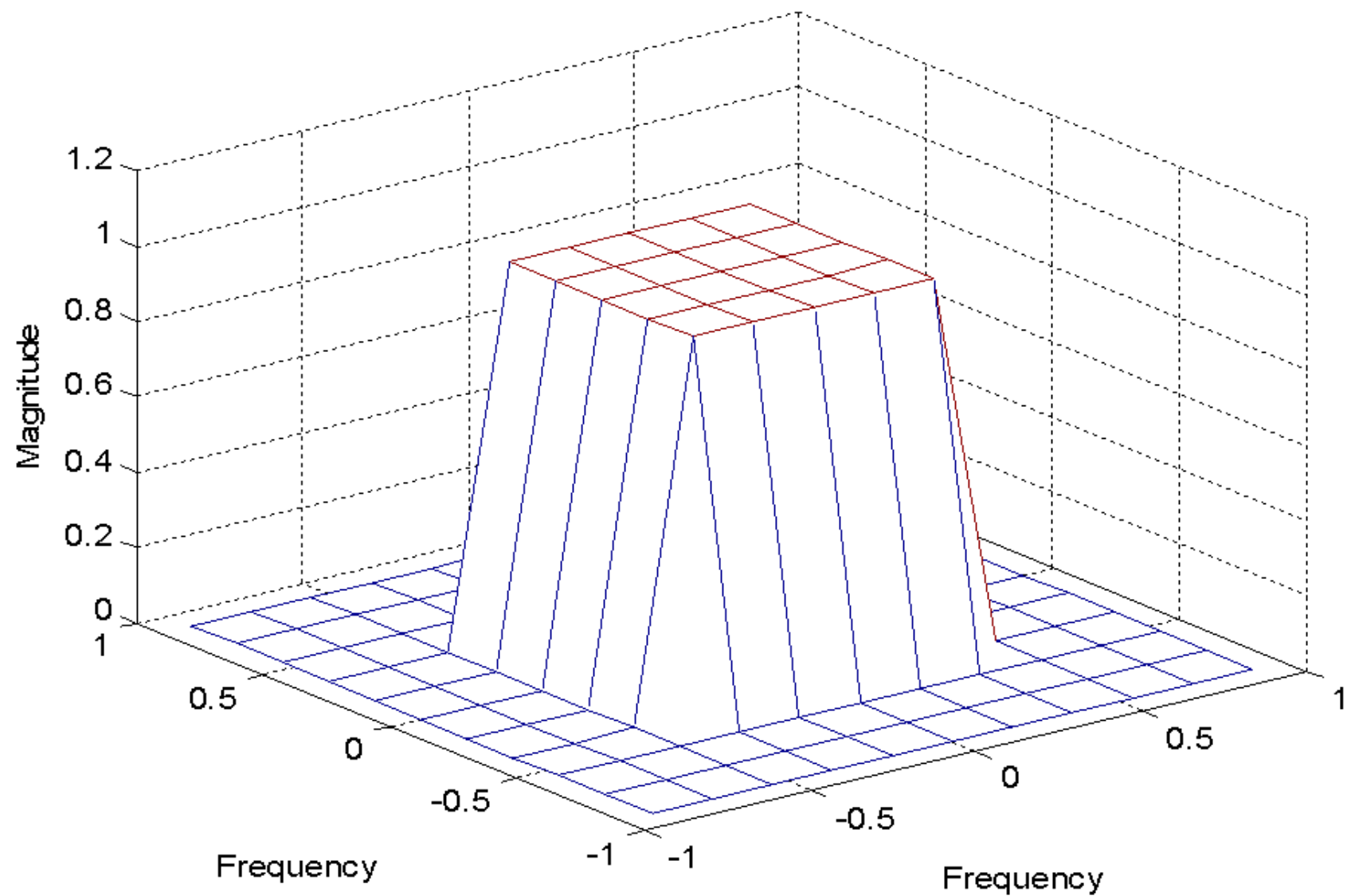
Ideal frequency response



Actual filter frequency response



Given filter ideal response



Hamming window smoothed filter

