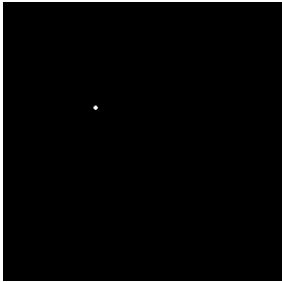
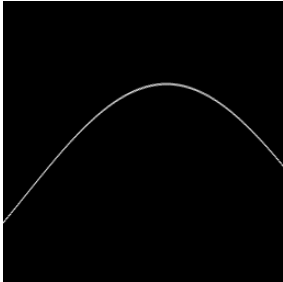
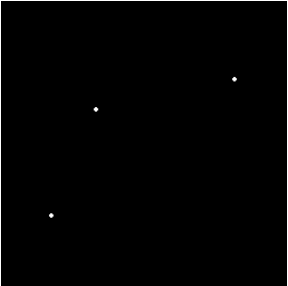
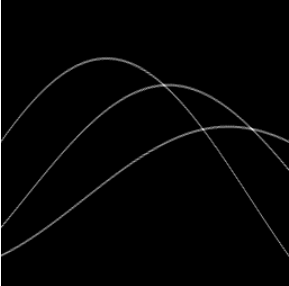
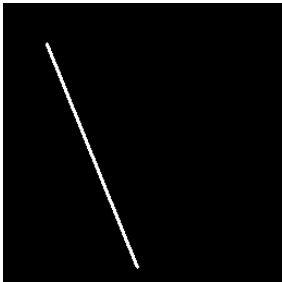
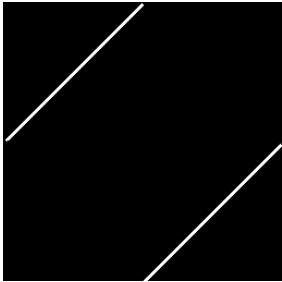


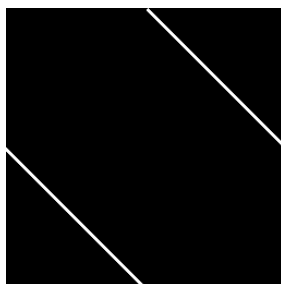
Skriptum

	Bsp.	Algorithmus
Canny	45, 46, 47, 48, 49	
Sobel	47	42
Hough-Transformation		50, 51
Medialachsen-Transformation	133 (B von E)	79
Gauss-Filter	34, 133 (G von A)	
Median-Filter	133 (J von A)	
Threshold	38, 134 (Q von P)	23
Fourier-Transformation	95, 96, 97, 98, 99	
Roberts		41
Prewitt		41

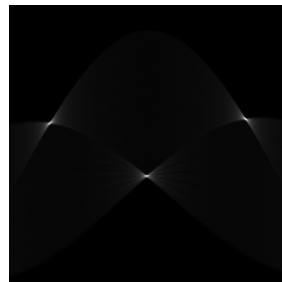
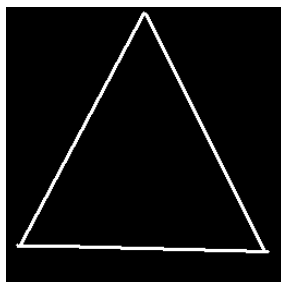
Hough-Transformation

	Image	Hough(Image)
Single Point		
Points		
Line		
Parallel Lines 1		

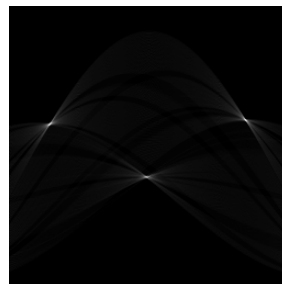
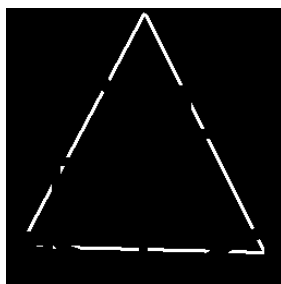
Parallel Lines 2



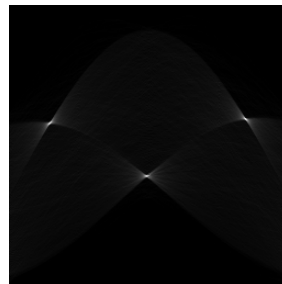
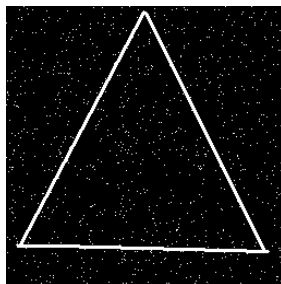
Triangle



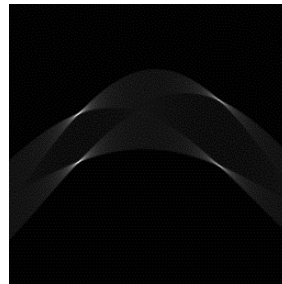
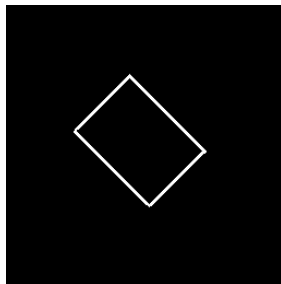
Triangle with absence



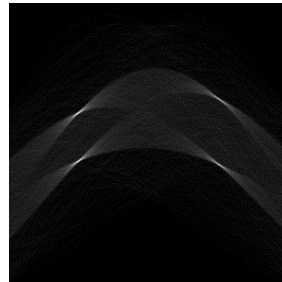
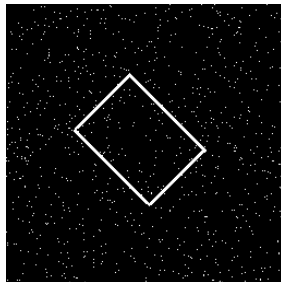
Triangle with noise



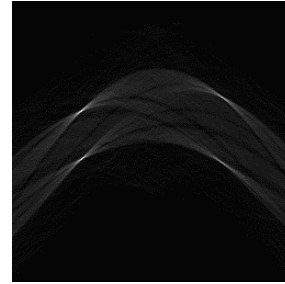
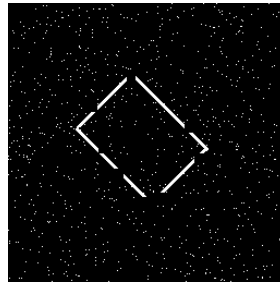
Rectangle



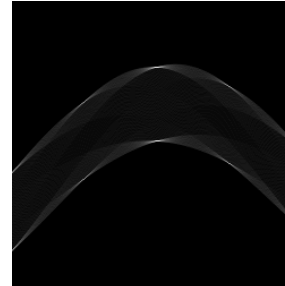
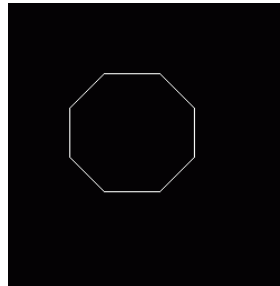
Rectangle with noise



**Rectangle with noise
and absence**



Octagon



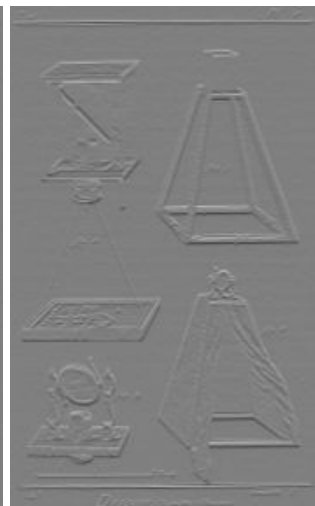
Sobel-Kantendetektor



Originalbild "Camera Obscura", das zur weiteren Berechnung genommen wurde.



Camera Obscura mit horizontalem Sobel gefaltet. Da auch negative Werte entstehen, wird der Nullpunkt als mittleres Grau dargestellt



Camera Obscura mit vertikalem Sobel gefaltet. Da auch negative Werte entstehen, wird der Nullpunkt als mittleres Grau dargestellt



Camera Obscura mit horizontalem und vertikalem Sobel gefaltet und kombiniert.

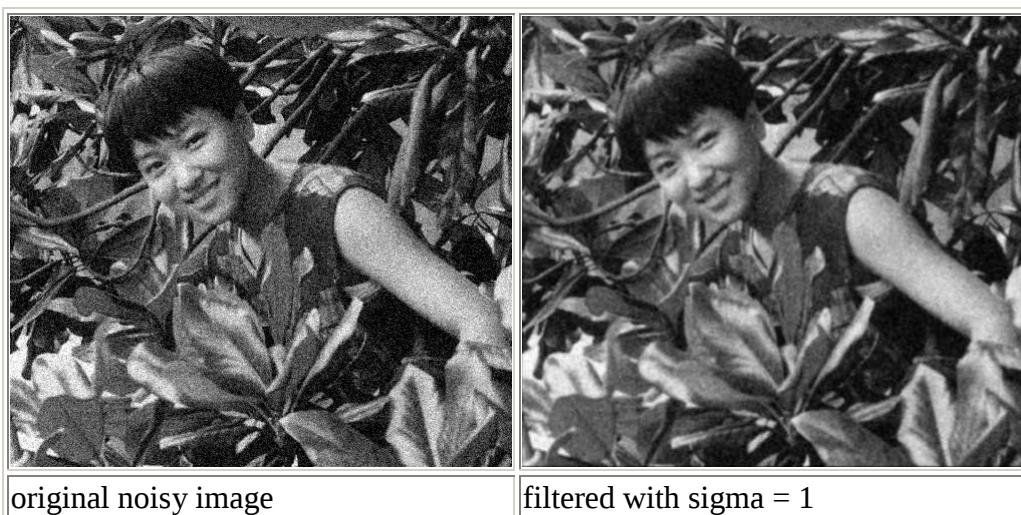
Gauss-Filter

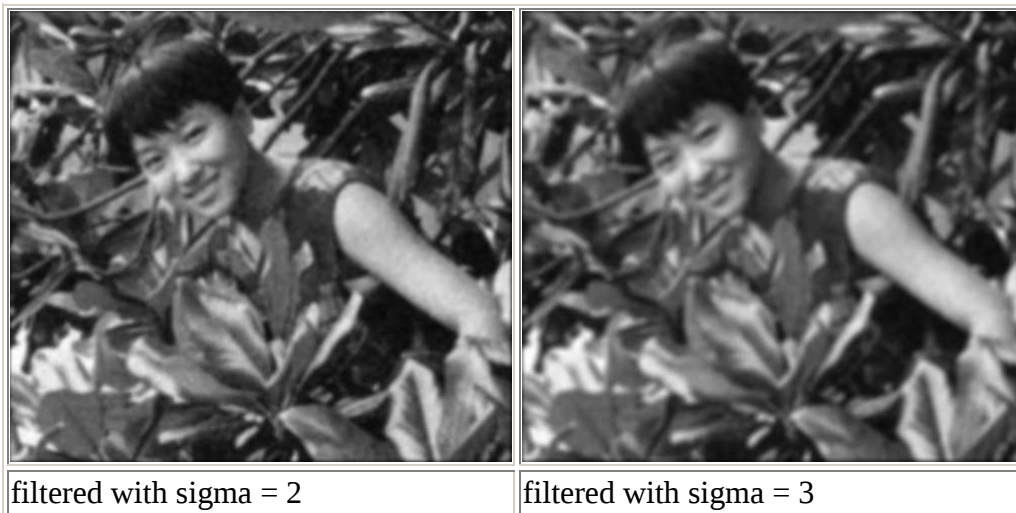
1. Smoothing Nonnoisy Image



2. Smoothing Gaussian-Noise Image

(The noise is generated by MATLAB function `imnoise(I,'gaussian',0,0.008)`)





3. Smoothing Salt and Pepper Noise Image

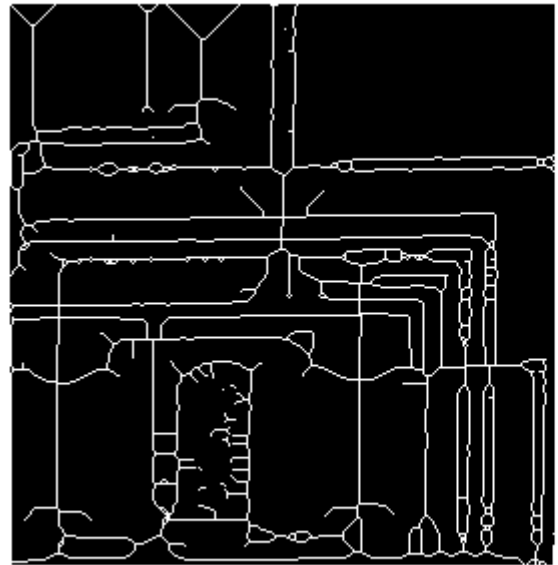
(The noise is generated by MATLAB function `imnoise(I,'salt & pepper',0.03)`)



`bwmorph(BW1,'skel',Inf);`



オリジナルイメージ



細線化されたイメージ

Fourier-Transformation

