

## Übung 5

## Rauschen

### Ziel dieser Übung

In dieser Übung werden verschiedene Varianten der Rauschunterdrückung, u.a. auch im Frequenzbereich, angewendet.

### Inhalte

- Statistische Rauschunterdrückung
- Rauschunterdrückung durch lokale Operatoren
- Rauschunterdrückung mit Hilfe der Fourier-Transformation
- Script-Programmierung in Matlab

## Theorie

### Bildrauschen

Rauschen ist eine allgemeine Bezeichnung für eine Vielzahl unerwünschter Störungen in einem System zur Bilderfassung. Diese Störungen können entweder *statisch* und damit zeitunabhängig sein, z.B. durch Unterschiede in den einzelnen Sensorzellen, oder sie können *dynamisch* sein und damit zeitlich variieren, wie z.B. beim **Temperaturrauschen**. Wenn das Rauschen alle Kanäle gleichermaßen betrifft, stellt es sich als **Helligkeitsrauschen** dar. Sind jedoch alle Farbkanäle unabhängig voneinander verrauscht, so spricht man von **Farbrauschen**.

然而，如果所有颜色通道都是独立的噪声

Typische Rauscharten und deren Ursachen sind:

#### **Temperaturrauschen:**

光感受器中的传导电子即使在 没有光入射的情况下也会移动，并 产生随机暗电流。 这一点随着温度的升高而变得更加强烈。

Die Leitungselektronen im Photorezeptor bewegen sich auch ohne Lichteinfall und erzeugen einen zufälligen Dunkelstrom. Dieser wird mit steigender Temperatur stärker.

#### **Photonenrauschen:**

Der auf den Sensor eintreffende Photonenstrom ist nicht gleichverteilt, sondern Poisson-verteilt. Der Abstand zwischen den Photonen ist zufällig, weshalb die tatsächlich eintreffende Anzahl der Photonen auch bei konstanter Belichtungszeit schwankt.

# Übung 5: Rauschen

Fachgebiet Industrielle Automatisierungstechnik

**Ausleserauschen:**

Bei höherer Auslesegeschwindigkeit entstehen Störungen beim Ladungstransport.

**Quantisierungsrauschen:**

Bei der Wandlung der analogen in diskrete Werte treten Rundungsfehler auf.

**Fixed Pattern Noise:**

Die einzelnen Photorezeptoren liefern produktionsbedingt leicht unterschiedliche Signale, weshalb ein statischer Versatz entsteht.

**Pixel Response Non Uniformity:**

Einzelne Pixel haben unterschiedliche Empfindlichkeitskurven.

Sensor- und Pixelgröße sind ausschlaggebende Faktoren für das Rauschen. Je größer die lichtempfindliche Fläche eines Bildpunkts ist, desto kleiner sind beispielsweise die Einflüsse des Photonenrauschens. Das Ausleserauschen wird auch deutlich verringert, wenn die Anzahl der Pixel (bei gleichbleibender Sensorfläche) kleiner ist.

Sensor- und Pixelgröße sind ausschlaggebende Faktoren für das Rauschen. Je größer die lichtempfindliche Fläche eines Bildpunkts ist, desto kleiner sind beispielsweise die Einflüsse des Photonenrauschens. Das Ausleserauschen wird auch deutlich verringert, wenn die Anzahl der Pixel (bei gleichbleibender Sensorfläche) kleiner ist.

## Qualität eines Signals

Als Maß für die Qualität eines Signals dient das *Signal-to-Noise Ratio* (SNR):

$$SNR = \frac{\text{Leistung des Nutzsignals}}{\text{Rauschleistung}} = \frac{P_{\text{signal}}}{P_{\text{noise}}}$$

Je größer das SNR, desto geringer ist der Störeinfluss des Rauschens.

## Modellierung des Rauschens

对加法和乘法噪声模型进行了区分。温度噪声是加性噪声的一个例子，因为暗电流与有用信号的强度无关。因此，该模型可表述如下。

Man unterscheidet zwischen dem **additiven und dem multiplikativen Rauschmodell**. Das Temperaturrauschen ist ein **Beispiel für additives Rauschen**, da der Dunkelstrom unabhängig von der Stärke des Nutzsignals ist. Das Modell kann also wie folgt formuliert werden:

$$g_{\text{verrauscht}} = g_{\text{original}} + r$$

与此相反的是乘法噪声。这里，噪声的强度取决于有用信号的强度。例如，设置较高的ISO值会同时增加信号和噪声，这就是为什么高ISO设置的夜间图像往往更有噪声。这里的噪声是乘法联系的。

Im Gegensatz dazu steht das multiplikative Rauschen. Hierbei ist die Stärke des Rauschens abhängig von der Stärke des Nutzsignals. Bei Einstellung eines größeren ISO Werts wird beispielsweise sowohl das Signal als auch das Rauschen verstärkt, weshalb Nachtbilder mit hoher ISO-Einstellung häufig stärker verrauscht sind. Das Rauschen ist hier multiplikativ verknüpft:

$$g_{\text{verrauscht}} = g_{\text{original}} \cdot (1 + r)$$

Nun wurde bestimmt, wie sich das Rauschen mit dem Nutzsignal zusammensetzt. Es ist aber noch unklar, wie das Rauschen selbst aussieht. Hierfür gibt es im Wesentlichen drei Modelle. Beim Gauß'schen Rauschen sind die Störungen um einen Mittelwert  $\mu$  mit einer Varianz  $\sigma^2$  normalverteilt. Darüber hinaus gibt es gleichverteiltes Rauschen und Impulsrauschen, auch Salt-and-Pepper Noise genannt. Bei letzterem sind vereinzelte Pixel entweder vollständig schwarz oder vollständig weiß. Dies kann z.B. durch „tote“ Pixel auf dem Sensor zustande kommen.

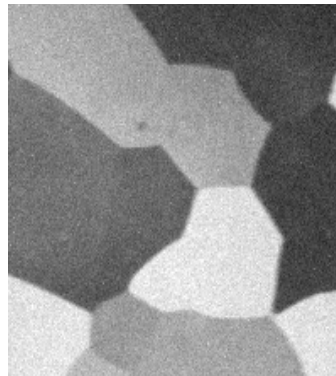
现在已经确定了噪声是如何与有用信号组成的。然而，目前仍不清楚噪声本身是什么样子。这方面基本上有三种模式。在高斯噪声下，干扰是围绕均值的高斯分布，方差为 $\sigma^2$ 。此外，还有平均分布的噪声和脉冲噪声，也称为盐和胡椒的噪声。在后两者中，孤立的像素要么是全黑，要么是全白。这可能是由传感器上的“死”像素造成的，例如。

# Übung 5: Rauschen

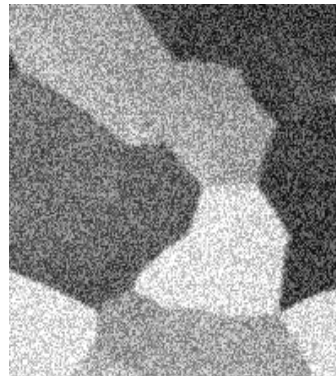
Fachgebiet Industrielle Automatisierungstechnik



Originalbild



Gauß'sches Rauschen



Gleichverteiltes Rauschen



Salt-and-Pepper Noise

## Rauschunterdrückung

在这个练习中，使用了三种不同的降噪方法。最简单的变体是几个录音的平均数。这里假定噪声是动态的，噪声成分的期望值为零。通过对足够多的图像进行平均化处理，尽管信噪比很差，但噪声还是可以被抑制。

In dieser Übung werden drei unterschiedliche Verfahren zur Rauschunterdrückung angewendet. Die einfachste Variante ist **die Mittelung mehrerer Aufnahmen**. Hierbei wird davon ausgegangen, dass das Rauschen dynamisch ist und der Erwartungswert des Rauschanteils gleich Null ist. Durch die Mittelung einer hinreichend großen Anzahl an Bildern kann das Rauschen trotz schlechtem SNR unterdrückt werden.

如果只有个别录音，必须使用另一种方法。例如，这些人可以是本地运营商。基本上，使用了三种不同的运算符。平均值过滤器、高斯过滤器和中值过滤器。前两个是卷积运算符（见卷积掩码），而后者是一个形态学运算符。尽管平均数和/或高斯滤波器对异常值，如发生在盐和胡椒噪声中的异常值有相当大的影响，但通常可以通过中值滤波器完全消除。

Stehen nur einzelne Aufnahmen zur Verfügung, so muss eine andere Methode angewendet werden. Das können beispielsweise **lokale Operatoren** sein. Im Wesentlichen werden drei verschiedene Operatoren angewendet: Mittelwertfilter, Gaußfilter und Medianfilter. Während die ersten beiden Faltungsoperatoren sind (siehe Faltungsmasken), ist letzterer ein morphologischer Operator. Während Ausreißer, wie sie z.B. beim Salt-and-Pepper Rauschen auftreten, trotz Mittelwert- und/oder Gaußfilter einen recht starken Einfluss haben, können sie vom Medianfilter in der Regel vollständig eliminiert werden.

1	1	1
1	1	1
1	1	1

3x3 Mittelwertfilter

1	4	1
4	12	4
1	4	1

3x3 Gaußfilter

此外，傅里叶变换也可用于抑制噪声。如果一个干扰的频率是已知的，它可以被专门消除。然而，通常情况下，噪声会对一个非常宽的频率范围产生影响。由于信号主要包含较低的频率，因此可以使用低通滤波器来去除大部分的噪声，而不会对图像产生太大影响。

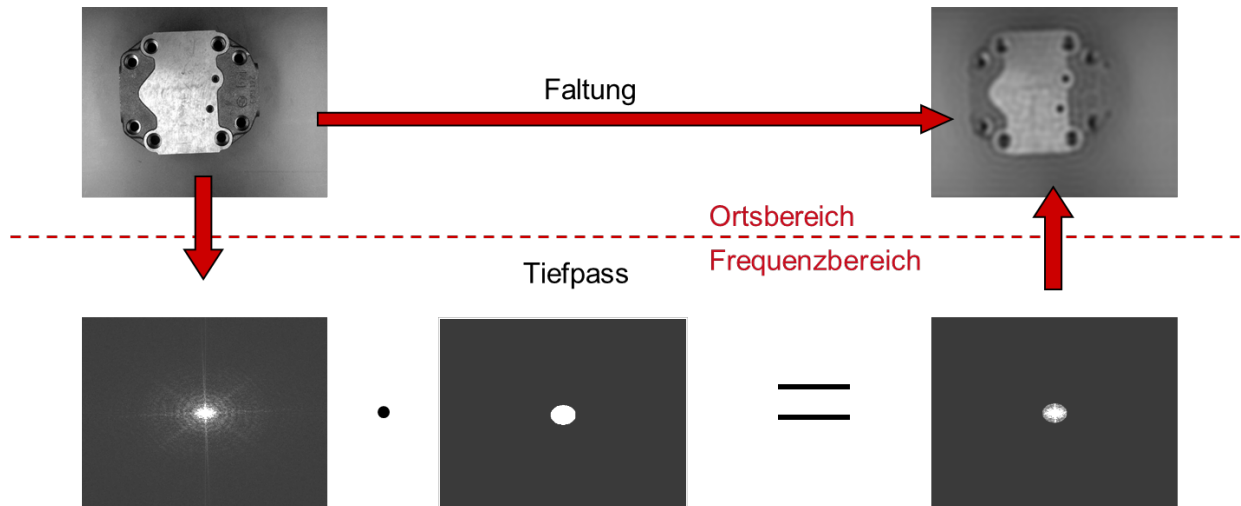
Weiterhin kann auch die **Fourier-Transformation** zur Rauschunterdrückung genutzt werden. Ist die Frequenz einer Störung bekannt, so kann diese gezielt eliminiert werden. Typischerweise wird das Rauschen aber Einfluss auf einen sehr breiten Frequenzbereich haben. Da das Signal hauptsächlich niedrigere Frequenzen enthält, kann über einen Tiefpass ein Großteil des Rauschens entfernt werden, ohne das Bild zu sehr zu beeinträchtigen.

Bei der Fourier-Transformation eines Bildes wird dieses in ein Amplitudenbild und ein Phasenbild zerlegt. Bildverarbeitungsoperatoren werden typischerweise auf das Amplitudenbild angewendet, das Phasenbild wird selten betrachtet. Im Amplitudenbild sind tiefe Frequenzen in der Mitte, je höher die Frequenz desto weiter am Rand wird

# Übung 5: Rauschen

Fachgebiet Industrielle Automatisierungstechnik

sie dargestellt. Um einen Tiefpass anzuwenden, wird das Amplitudenbild also mit einer Maske multipliziert, die die Werte in der Mitte (die tiefen Frequenzen) beibehält und alle anderen Werte gleich Null setzt. Anschließend wird das so veränderte Amplitudenbild gemeinsam mit dem Phasenbild wieder zurück transformiert. Eine im Frequenzbereich durchgeführte Multiplikation entspricht einer Faltung auf dem Originalbild:



## Aufgaben

### 1. Rauschunterdrückung durch Mittelung

Es gibt zwei verschiedene Bildersätze mit jeweils 100 Aufnahmen von einem Pumpengehäuse (Bildrauschen1\_#.png, Bildrauschen2\_#.png). Die einzelnen Bilder sind leider stark verrauscht.

- Führen Sie im Matlab LiveScript eine Rauschunterdrückung durch, indem Sie die Bilder beider Serien jeweils mitteln! Innerhalb einer for-Schleife soll in jedem Schleifendurchlauf ein Bild der Serie eingelesen und in uint16 gewandelt werden und diese Bilder aufsummiert werden (`imadd`). Nutzen Sie die Matlab-Hilfe um sich über diese Funktion zu informieren!

- Unter welchen Bedingungen führt die Rauschunterdrückung durch Mittelung zum Erfolg? Diskutieren Sie, welche Vor- und Nachteile dieses Verfahren in der Praxis haben könnte!

### 2. Rauschunterdrückung durch lokale Operatoren

Sie haben drei einzelne Aufnahmen des Pumpengehäuses, die mit jeweils unterschiedlichen Arten von Rauschen behaftet sind (*Bild1*, *Bild2*, *Bild3*). Diese sollen nun mit Hilfe von Mittelwert-, Gauß- und Medianfilter verbessert werden.

- Welche Arten von Rauschen können Sie auf den drei Bildern erkennen? Diskutieren Sie, welche Filter für welches Rauschen am besten geeignet sein könnten!
- Überprüfen Sie Ihre Vermutungen nun experimentell. Lesen Sie im Script die einzelnen Bilder ein und wenden Sie die verfügbaren Filter darauf an. Benutzen Sie für Mittelwertfilter `fspecial`, für den Gaußfilter `imgaussfilt` und für den Medianfilter die Funktion `medfilt2`.

- Beurteilen Sie nach Ihrem subjektiven Empfinden die Ergebnisbilder. Inwieweit wurden Ihre Vermutungen bestätigt oder widerlegt? Inwiefern unterscheiden sich die verschiedenen Filter bzgl. der Erhaltung von tatsächlichen Kanten im Bild?

### 3. Rauschunterdrückung im Frequenzbereich

Die Rauschunterdrückung soll nun nicht mehr über lokale Operatoren, sondern global im Frequenzbereich durchgeführt werden. Grundsätzlich entspricht eine Faltung einer Multiplikation im Frequenzbereich, sodass wir die lokalen Operatoren aus Aufgabe 2 (Kernels) zu Masken im Frequenzbereich umwandeln können und auf das Amplitudenbild anwenden können. Zusätzlich dazu können wir aber auch andere Operationen im Frequenzbereich durchführen. Dazu gehören u.a. der Hochpass zur Kantendetektion und der Tiefpass zur Rauschunterdrückung.

# Übung 5: Rauschen

Fachgebiet Industrielle Automatisierungstechnik

- Entwickeln Sie ein Script, das ein Bild, z.B. "verschmutztes\_Werkstuek\_FhG\_IPK.jpg", einliest, in den Fourierbereich transformiert und auch wieder zurück. Lassen Sie sich alle Zwischenergebnisse anzeigen und probieren Sie verschiedene Bilder aus, um das Amplitudenbild interpretieren zu können. Verwenden Sie für die Transformation `fft2` bzw. `ifft2`. Um das Amplitudenbild sichtbar zu machen, eignet sich `imshow(log(abs(FFTImage) + 1), [])`.
- Um die niedrigen Frequenzanteile in der Bildmitte statt in den Außenbereichen anzuzeigen, verwenden Sie im Script `fftshift`.
- Fügen Sie nun die eigentliche Operation (Hoch-/Tiefpass) im Frequenzbereich ein, indem Sie das Frequenzbild mit einer Filtermaske multiplizieren. Zum Erstellen der Filtermaske verwenden Sie die vorbereitete Funktion `createFFTMask(imageSizeX, imageSizeY, radius, "highpass/lowpass")`. Anschließend müssen die niedrigen Frequenzanteile mit `ifftshift` wieder aus dem Bildzentrum in die Bildecken verschoben werden und das Bild mit `ifft2` zurücktransformiert werden. Lassen Sie sich die Amplitudenbilder vor und nach der Multiplikation anzeigen, um die Operation besser nachvollziehen zu können.
- Wenden Sie Ihr Skript für Hoch- und Tiefpass auch auf die Testbilder *blackTriangle*, *whiteblock*, *whiteCircle* an. Sie werden Filterungs-Artefakte beobachten (sog. „Ringing“). Recherchieren Sie, warum diese Artefakte entstehen und welche Möglichkeiten es gibt um deren Entstehung zu verhindern!

Bild einlesen und in *grayscale* konvertieren:

Fourier-Transformation:

niedrige Frequenzanteile in die Bildmitte:

Filtermaske erstellen und mit transformiertem Bild multiplizieren (mit `.*` Array multiply):

niedrige Frequenzanteile mit `ifftshift` wieder aus dem Bildzentrum in die Bildecken:

inverse Fourier-Transformation und gefiltertes Bild anzeigen: