

# Intelligente Sehsysteme

## **4 Nichtlineare Filter und Filtern von Farbbildern**

---

Rangordnungsfiler, Medianfilter, Diffusionsfilter

RGB- und HSI-Farbmodelle,

Glättung und Kantenhervorhebung von RGB- und HSI-Farbbildern

*Henry Hölzemann*

# Inhalt

---

- Nichtlineare Filter
  - Rangordnungsfilter
    - Medianfilter
      - Kantenerhaltung beim Medianfilter
  - Diffusionsfilter
    - isotrope homogene Diffusion
    - isotrope inhomogene Diffusion
    - anisotrope Diffusion
- Filterung von Farbbildern
  - Glättung von Farbbildern
  - Kanten hervorhebung in Farbbildern

# Rückblick auf lineare Filterung

---

- Kontrastverbesserung durch **lineare Filterung** erfolgt durch
  - **Unterdrückung von Störsignalen**  
(Mittelwertfilter, Gauß-Filter, Binomialfilter)
  - **Hervorhebung/Extraktion von Kantensignalen**  
(Sobel-Operator, Laplace-Operator, LoG-Operator, DoG-Operator)
- **LEIDER** führen lineare Operatoren zur **Rauschunterdrückung** auch zur **Kantenunterdrückung**.
- **DAHER** muss i. A. eine bzgl. der Anwendung **angemessene Balance** zwischen **Rauschunterdrückung** und **Kantenhervorhebung** gewählt werden.

# Nichtlineare Filterung (1)

---

- Die bei linearen Filtern notwendige Balance zwischen Rauschunterdrückung und Kantenhervorhebung ist für viele Anwendungen akzeptabel, da diese Balance hinsichtlich der Anwendung angepasst werden kann.
- Entsprechend gehören lineare Filter zum Standardmethodenarsenal des Computersehens und insbes. der Low-Level-Vision bzw. der Bildverarbeitung.

## Nichtlineare Filterung (2)

---

- Für schwierige Anwendungsfälle kann aber eine effizientere Trennung zwischen Rauschunterdrückung und Kanten hervorhebung notwendig sein.
- Zwei Filterklassen können diese Trennung zwischen Rauschunterdrückung und Kanten hervorhebung effizienter gestalten als lineare Filter:
  - Rangordnungsfiler
  - Diffusionsfilter
- Beide Filterklassen gehören zur Klasse der nichtlinearen Filter.

# Rangordnungsfilter (1)

---

## Rangordnungsfilter

- werden **wie lineare Filter** auf ein Pixel  $p$  und seine Nachbarpixel angewandt – wir gehen hier zunächst von Nachbarpixeln innerhalb einer um  $p$  zentrierten quadratischen Umgebung *ungerader Größen* ( $3 \times 3$ ,  $5 \times 5$ , usw.) aus.
- berechnen im **Gegensatz zu linearen Filtern** keine gewichteten Summen über den Intensitäten innerhalb der Umgebung, sondern
  - **sortieren alle Pixel** innerhalb der Umgebung *gemäß ihrer Intensitätswerte*
  - **liefern als Ergebnis den an einem bestimmten Rang** der sortierten Reihenfolge **eingeordneten Wert**

## Rangordnungsfiler (2)

---

- Die Anzahl der Operationen ist für lineare Filter linear in der Anzahl der Pixel der quadratischen Umgebung. Die Zeitkomplexität ist also  $O(n)$  für  $n = m^2$  und Größe  $m$  des Filterkerns.
- Die Zeitkomplexität für Rangordnungsfiler ist wegen der Sortierung dagegen  $O(n \log n)$  für  $n = m^2$  und Größe  $m$  des Filterkerns.
- Für kleine Filterkerne ist dies nicht relevant. Für sehr große Filterkerne und sehr große Bilder kann es zu spürbaren Unterschieden in den Berechnungszeiten kommen.\*

---

\* Zumal sich für lineare Filter mit großen Filterkernen noch die FFT als effizienterer Weg anbietet. Die Konvolution eines Bildes der Größe  $N \times N$  mit einem Faltungskern der Größe  $m \times m$  hat die Zeitkomplexität  $O(m^2 \cdot N^2)$ . Über FFT ausgeführt ist sie  $O(N^2 \cdot \log_2 N)$ . Für  $\log_2 N < m^2$  ist die Konvolution also über FFT effizienter ausführbar.

# Medianfilter (1)

---

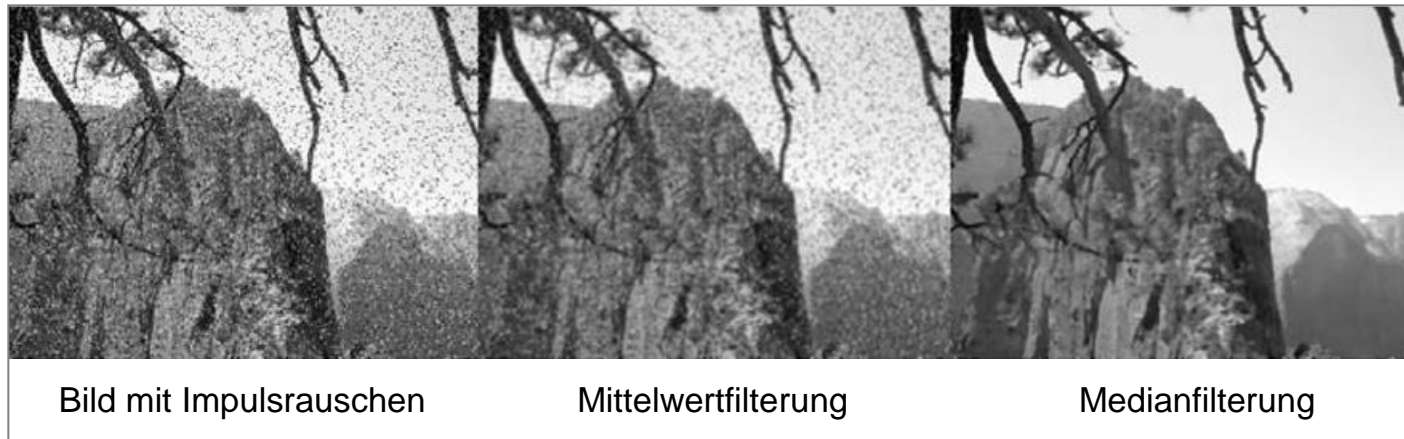
- Der **Median** ist der Wert in der mittleren Position einer sortierten Folge.
- Das **Medianfilter** ist ein Rangordnungsfilter, das als Ergebnis den Median liefert
- Median und Mittelwert einer sortierten Wertefolge sind i. A. ungleich, nähern sich aber für große Folgen normalverteilter Werte einander an
- Daraus folgt die Brauchbarkeit des Medianfilters für die Rauschunterdrückung
- Ist die ungestörte Bildfunktion  $I_u(x,y)$  in der Filterumgebung konstant und der Erwartungswert des Rauschens Null, so entspricht der durch den Median angenäherte Erwartungswert der gesuchten ungestörten Bildfunktion.
- Ähnlich wie beim Mittelwertfilter wird die Näherung bei steigender Filtergröße des Medianfilters besser.



## Medianfilter (2)

---

- Das Impulsrauschen (Salt-and-Pepper-Noise) ist ein Spezialfall des Rauschens
- Für das lineare Mittelwertfilter wären sehr große Bereiche nötig, um den Erwartungswert der ungestörten Bildfunktion anzunähern
- Da aber der Großteil der Intensitätswerte beim Impulsrauschen ungestört ist und damit dem Erwartungswert der ungestörten Bildfunktion entspricht, führt die Medianfilterung in homogenen Bereichen zur vollständigen Beseitigung des Impulsrauschens:



# Kantenerhaltung beim Medianfilter (1)

---

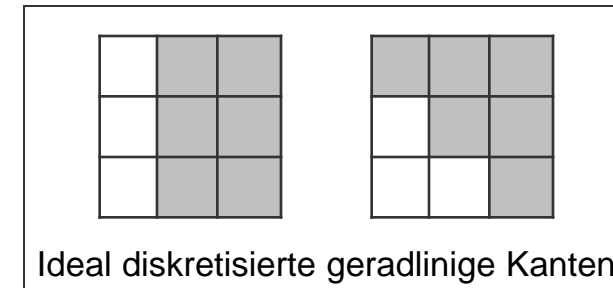
Unter den folgenden Umständen ist das Medianfilter kantenerhaltend:

- 1) die Intensitätswerte der ungestörten Bildfunktion  $I_u(x,y)$   
sind auf beiden Seiten der Kante jeweils *konstant*
- 2) der Signalabstand, d.h. der Werteunterschied an einer Kante,  
ist größer als die Rauschamplitude
- 3) die Kante verläuft über die Fläche des Medianfilters gerade

# Kantenerhaltung beim Medianfilter (2)

Diskussion:

- Das Medianfilter werde an einer Kante  $k$  angewandt
- In der sortierten Folge der Pixelintensitäten bezeichnen  $M_D$  bzw.  $M_H$  die Mengen der Pixel auf der dunkleren bzw. helleren Seite der Kante  $k$
- Gilt Bedingung 1 (konstante Werte auf jeder Seite von  $k$ ), werden alle Werte aus  $M_D$  vor allen Werten aus  $M_H$  einsortiert
- Gilt Bedingung 2 (Kantensignal stärker als Rauschen), bleibt die o.g. Sortierung der Werte aus  $M_D$  und  $M_H$  auch bei überlagertem Rauschen erhalten
- Gilt Bedingung 3 (gerade Kante), so umfasst  $M_D$  mehr Pixel als  $M_H$ , wenn das zentrale Pixel zu  $M_D$  gehört – und umgekehrt



Bemerkung: das Medianfilter ist an einer Kante nicht mehr länger eine gute Näherung des Mittelwertfilters der Pixel einer homogenen Region, da er über Pixel aus zwei unterschiedl. Regionen gebildet wird. Aber genau das passt!

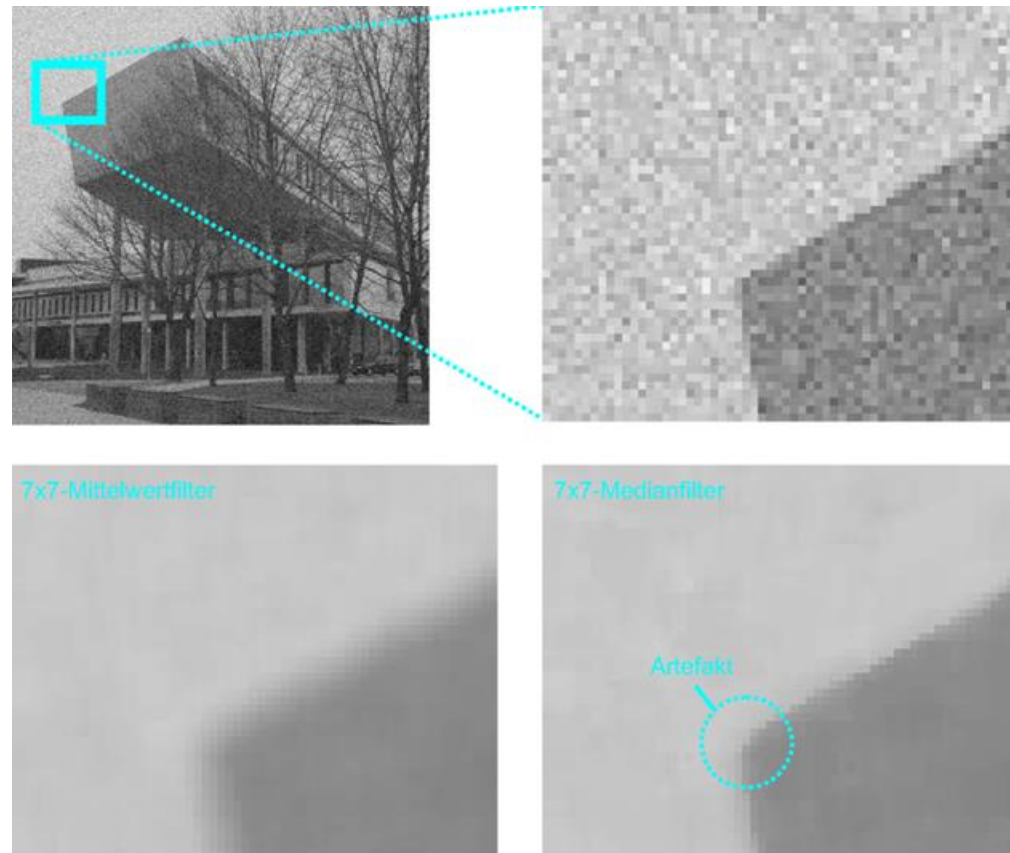
## Kantenerhaltung beim Medianfilter (3)

---

- Von den drei Bedingungen ist die 3. Bedingungen für „Realweltszenen“ am schwierigsten einzuhalten:
  - Kantenzüge verlaufen selten über längere Strecken geradlinig!
- Wird das Medianfilter jedoch klein genug gewählt, so wird die 3. Bedingungen jedoch in den meisten Bildbereichen angenähert.
  - Die einzige **Ausnahme** sind dann noch **Eckpunkte**:
    - So klein der Filterkern auch gewählt wird, der Kantenverlauf ist an einer Ecke per Definition nicht geradlinig!
    - Somit verursacht das Medianfilter zunächst Artefakte an Ecken.

# Kantenerhaltung beim Medianfilter (4)

Das Medianfilter erhält Kanten nur, wenn sie innerhalb der Filterumgebung nahezu gerade verlaufen. Das ist an Ecken nicht der Fall:

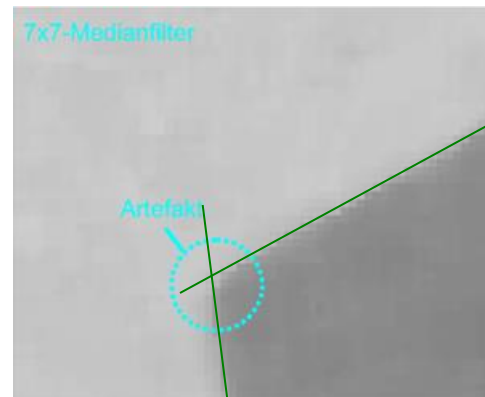


# Kantenerhaltung beim Medianfilter (5)

---

Die vom Medianfilter an Ecken erzeugten Artefakte sind ggf. (d.h. je nach Interpretation der Artefakte und nach Anwendung) aber auch kompensierbar:

- Wird das Artefakte als Positionsverschiebung der tatsächlich abgebildeten Objektecke interpretiert,
- so kann diese Eckenposition ggf. wieder durch die **Schnittbildung der extrahierten Kanten** zurück gewonnen werden:



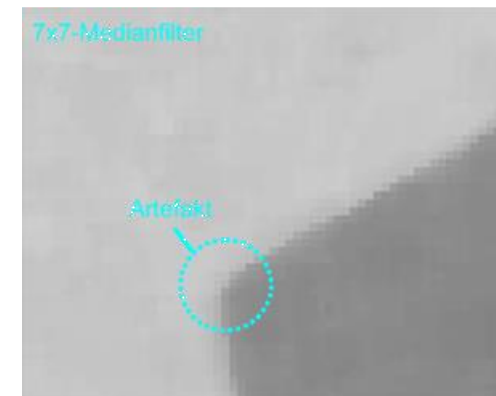
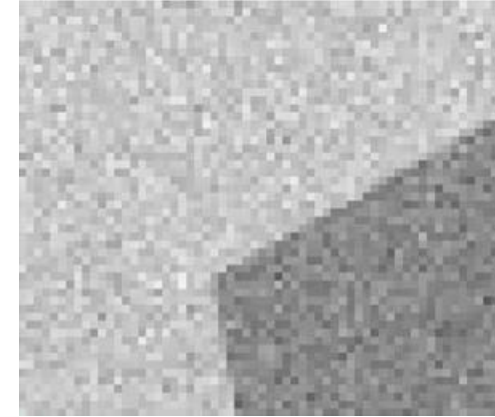
# Kantenerhaltung beim Medianfilter (6)

Das Medianfilter wirkt also

- rauschunterdrückend in homogenen Bereichen und
- kantenerhaltend an Kanten.

Die Größe des Medianfilters sollte also eine Balance sein zwischen

- möglichst guter Annäherung an den Erwartungswert der ungestörten Bildfunktion  $I_u(x,y)$  durch große Filter und
- einer Maximierung der Anzahl der Kantenpixel, die zu lokal geradlinig verlaufenden Kanten gehören.

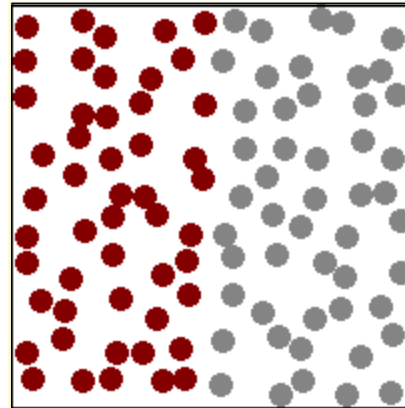


Bildquelle: Klaus Tönnies: Grundlagen der Bildverarbeitung, Pearson Studium, 2005.

# Diffusionsfilter (1)

---

Die Idee der Diffusionsfilter basiert auf der [Simulation des physikalischen Diffusionsprozesses](#):





## Diffusionsfilter (2)

---

Die Idee der Diffusionsfilter basiert auf der [Simulation des physikalischen Diffusionsprozesses](#):

- die Bildfunktion  $I(x,y)$  wird als Konzentration einer Flüssigkeit interpretiert
- der Diffusionsprozess gleicht Konzentrationsvariationen, also eine inhomogen verteilte Konzentration, über die Zeit aus

Der Ansatz der Diffusionsfilter stammt von Perona und Malik (1990)\*

---

\* P. Perona, J. Malik (1990): Scale-Space and Edge Detection Using Anisotropic Diffusion. IEEE PAMI, 12(7).

## Diffusionsfilter (3)

---

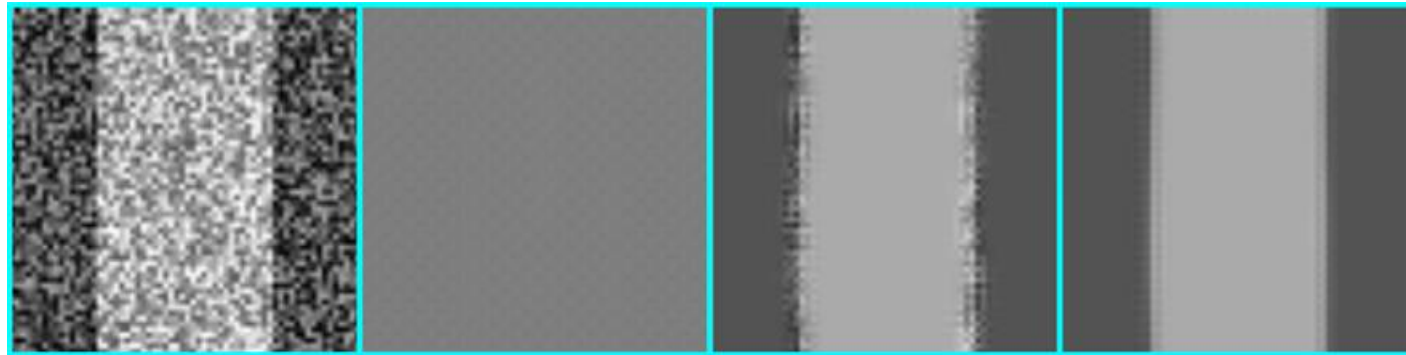
Es gibt drei verschiedene Diffusionsmodelle:

- 1) **Isotrope homogene Diffusion** führt den gleichmäßigen Ausgleich in alle Richtungen durch
- 2) **Isotrope inhomogene Diffusion** macht das Ausmaß der Diffusion vom Vorhandensein einer Kante in der Nachbarschaft ab
- 3) **Anisotrope Diffusion** findet auch an Kanten statt, aber nur parallel zur Kantenrichtung und nicht über die Kante hinaus

## Diffusionsfilter (3)

Beispiel für die drei verschiedene Diffusionsmodelle (nach 500 Iterationen) an einem verrauschten Bild (links mit  $SNR_{max} = 0,25:1$ ) :

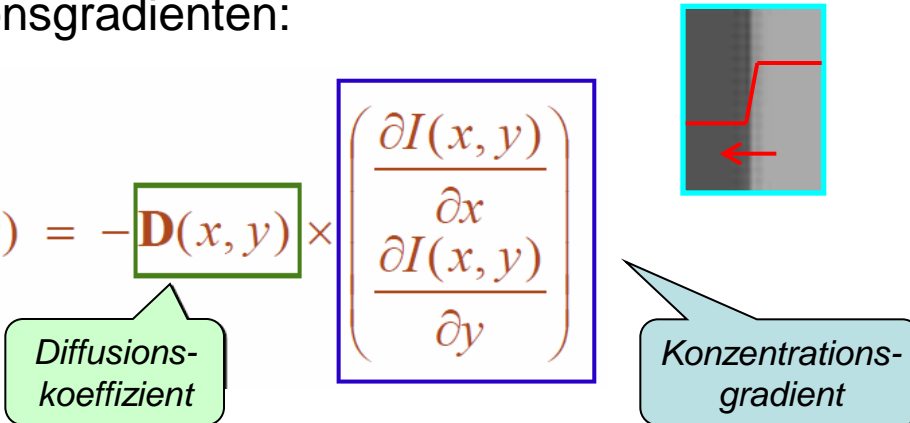
- 1) Die **isotrope homogene Diffusion** löscht alle Kanten vollständig.
- 2) Die **isotrope inhomogene Diffusion** erhält den Unterschied zw. Vordergrund und Hintergrund durch Kantenbegrenzung, zeigt aber verrauschte Kanten.
- 3) Die **anisotrope Diffusion** erfolgt nur parallel zur Kantenrichtung und nicht über die Kante hinaus. Daher bleibt der Unterschied zwischen Vordergrund und Hintergrund erhalten *und* das Rauschen an den Kanten wird unterdrückt.



## Diffusionsfilter (4)

- Die Diffusionsrate ist von einem *Diffusionskoeffizienten*  $D$  abhängig, der sich im mehrdimensionalen Fall als Diffusionstensor  $\mathbf{D}$  darstellt.\*
- Der *Diffusionstensor*  $\mathbf{D}$  spezifiziert die Diffusionsrate in Abhängigkeit von der Richtung eines *Konzentrationsgradienten*  $\nabla u$ .
- Soll durch die Diffusion das Rauschen in Bildern unterdrückt werden, so ist der Konzentrationsgradient der Intensitätsgradient der Bildfunktion  $I(x,y)$  und der Diffusionstensor  $\mathbf{D}$  eine  $2 \times 2$ -Matrix.
- Nach dem 1. *Fickschen Gesetz* ist der *Fluss*  $\vec{j}$  (bzw. die *Teilchenstromdichte*) proportional zum negativen Konzentrationsgradienten:

$$(1) \quad \vec{j}(x, y) = -\mathbf{D}(x, y) \times \nabla u(x, y) = -\mathbf{D}(x, y) \times \begin{pmatrix} \frac{\partial I(x, y)}{\partial x} \\ \frac{\partial I(x, y)}{\partial y} \end{pmatrix}$$



\* Ein Tensor ist eine multilineare Abbildung, d.h. eine Abbildung, die in jeder Variablen linear ist.

## Diffusionsfilter (5)

---

- Der Fluss  $\vec{j}(x,y,t)$  ändert sich über die Zeit  $t$  und verursacht zur Zeit  $t$  an der Stelle  $(x,y)$  eine Konzentrationsänderung:

$$\frac{\partial u(x,y,t)}{\partial t} = -\operatorname{div} \vec{j}(x,y,t) \quad (2)$$

- Der Divergenzoperator *div*
  - ist ein Differentialoperator
  - beschreibt anschaulich für jeden Raumpunkt  $\mathbf{p}$ :  
„wie viel mehr fließt aus einer Umgebung von  $\mathbf{p}$  hinaus als in sie hinein“
    - ~ Quellen zeigen eine Divergenz größer als Null
    - ~ Senken zeigen eine Divergenz kleiner als Null hat
    - ~ Bei Divergenz gleich Null bezeichnet man das Feld als *quellenfrei*

## Diffusionsfilter (6)

---

- Genauer: die Divergenz des Flusses  $\vec{j}$  an einem Punkt  $\mathbf{p}$  ist der Grenzwert des Flusses über den Rand  $\rho(R)$  einer den Punkt  $\mathbf{p}$  umgebenden Region  $R$  (normiert durch die Größe  $|R|$  von  $R$ ) für das Schrumpfen von  $R$  auf den Punkt  $\mathbf{p}$ :

$$\operatorname{div} \vec{j}(\mathbf{p}) = \lim_{|R| \rightarrow 0} \left( \frac{1}{|R|} \int_{\rho(R)} \vec{j} \cdot \vec{n} \, dS \right). \quad (3)$$

$R$  ist eine beliebige Region (z.B. eine 3D-Kugel) mit Volumen  $|R|$ . Die Integration erfolgt über den Rand  $\rho(R)$  von  $R$ . Vektor  $\vec{n}$  ist die nach außen gerichtete Normale und  $dS$  das zugehörige Flächenelement.

- Bei Spezialisierung auf infinitesimale Würfel erhält man eine Darstellung in kartesischen Koordinaten – hier 2-dimensional mit  $(x, y, t)$  für  $\mathbf{p}$ :

$$\operatorname{div} \vec{j}(x, y, t) = \left( \frac{\partial j_x(x, y, t)}{\partial x} + \frac{\partial j_y(x, y, t)}{\partial y} \right). \quad (4)$$

Dabei sind  $j_x$  und  $j_y$  die beiden Richtungskomponenten des Flussvektors.

# Diffusionsfilter (7)

---

Die **Simulation** des Diffusionsprozesses

- basiert auf sukzessiven Berechnungen von Konzentrationen  $u(x,y,t)$  für Zeiten  $t$
- ausgehend von einer Anfangskonzentration  $u(x,y,0)$ , die dem zu bearbeitenden Bild entspricht:  $u(x,y,0) = I(x,y)$ .

**ABER:** für beliebige Diffusionstensoren ist dies nicht analytisch lösbar.

Daher: *iterative Schätzung* der Konzentration zur Zeit  $t_{i+1}$  aus der zur Zeit  $t_i$ :

$$u(x, y, t_{i+1}) = u(x, y, t_i) + (t_{i+1} - t_i) \frac{\partial u(x, y, t_i)}{\partial t}. \quad (5)$$

Die Schätzung geht von einer über dem Zeitintervall  $(t_{i+1} - t_i)$  konstanten Rate  $\partial u / \partial t$  aus und wird nur für Positionen  $(x,y)$  auf dem Pixelraster geschätzt.

# Zusammenfassung: Alle Schritte der Diffusionsfilter

$$\begin{pmatrix} 1 \\ 0 \\ -1 \end{pmatrix} \begin{pmatrix} -1 & 0 & 1 \end{pmatrix}$$

- 1) Approximiere Intensitätsgradienten  $\partial I(x,y,t)/\partial x$  und  $\partial I(x,y,t)/\partial y$  durch Differenzen der Intensitäten in x- bzw. y-Richtung, z.B.:  $I(x+1,y) - I(x-1,y)$ ,  $I(x,y+1) - I(x,y-1)$

- 2) Berechne **Diffusionstensor D**

Noch zu klären!  
Abhängig vom Modell des Diffusionsfilters

- 3) Rechne Fluss  $\vec{j}(x,y,t) = (j_x(x,y,t), j_y(x,y,t)) = -\mathbf{D}(x,y,t) \times \begin{pmatrix} \frac{\partial I(x,y)}{\partial x} \\ \frac{\partial I(x,y)}{\partial y} \end{pmatrix}$  (1)

- 4) Approximiere Flussgradienten  $\partial j_x(x,y,t)/\partial x$  und  $\partial j_y(x,y,t)/\partial y$  durch Differenzen des Flusses in x- bzw. y-Richtung, z.B.:  $j_x(x+1,y) - j_x(x-1,y)$ ,  $j_y(x,y+1) - j_y(x,y-1)$

- 5) Rechne Divergenz  $\text{div } j(x,y,t) = \partial j_x(x,y,t)/\partial x + \partial j_y(x,y,t)/\partial y$  (4)

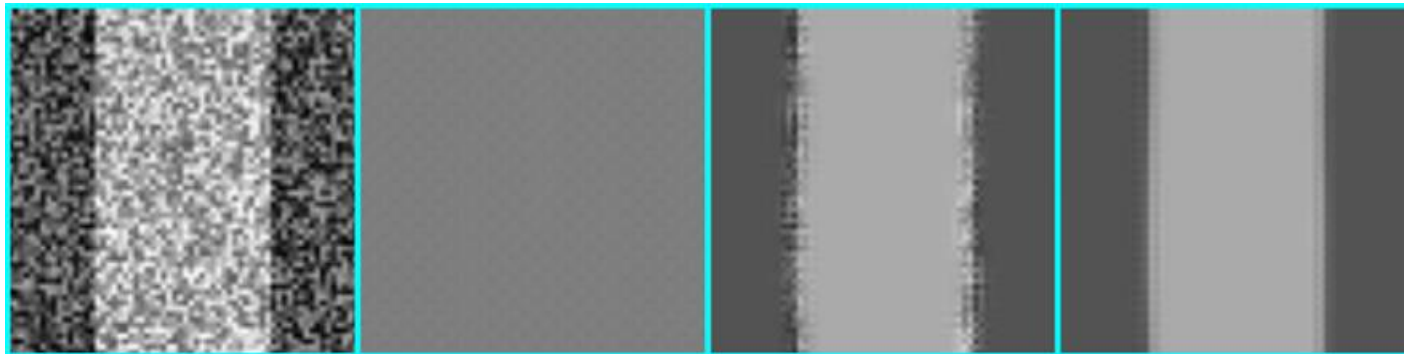
- 6)  $I(x,y,t+1) = I(x,y,t) - \text{div } j(x,y,t) = I(x,y,t) - (\partial j_x(x,y,t)/\partial x + \partial j_y(x,y,t)/\partial y)$  (2),(5)




# Diffusionsmodelle

---

- Verschied. Diffusionsfilter unterscheiden sich durch die Diffusionstensoren.
- Wir betrachten die drei genannten Diffusionsmodelle:
  - 1) **isotrope homogene Diffusion** mit gleichmäß. Ausgleich in alle Richtungen
  - 2) **isotrope inhomogene Diffusion** mit unterschiedl. Ausmaß der Diffusion
  - 3) **anisotrope Diffusion**, die nur parallel zur Kantenrichtung und nicht über Kante hinaus erfolgt



# Isotrope homogene Diffusion (1)

  $\begin{pmatrix} \varepsilon_0 & 0 \\ 0 & \varepsilon_0 \end{pmatrix}$

- Für die isotrope homogene Diffusion (Variante 1) ist **D** die mit einem Skalar  $\varepsilon_0$  multiplizierte Identitätsmatrix. Die Intensitätsveränderung  $\partial u / \partial t$  ist also

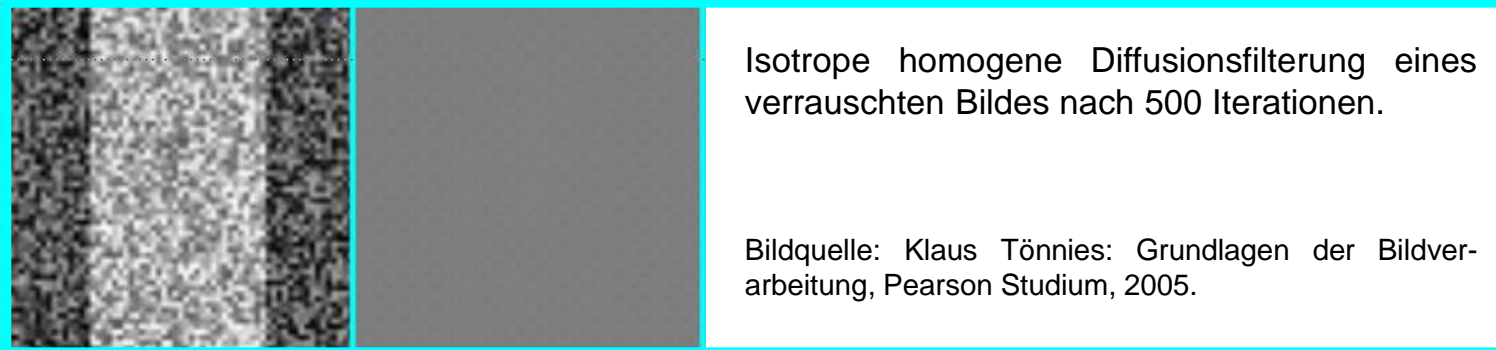
$$\partial u(x, y, t) / \partial t = \varepsilon_0 (\partial u / \partial x + \partial u / \partial y). \quad (6)$$

- Die Intensitätsveränderung erfolgt also über einen konstanten Tensor und hängt nur vom lokalen Intensitätsgradienten ab!
- Hierfür gibt es eine analytische Lösung:
  - ↪ danach ist die Diffusion einer Anfangsverteilung  $I(x, y)$  nach der Zeit  $t$  das Ergebnis einer Konvolution mit einer Gauß-Funktion mit  $\sigma = (2t)^{1/2}$ .

## Isotrope homogene Diffusion (2)

---

- Erwartungsgemäß führt die isotrope homogene Diffusion daher bei hinreichender Zahl von Iterationen zum vollständigen Ausgleich der Intensitätswerte.
- Das Resultat ist ein geglättetes Bild mit unscharfen (ggf. nicht mehr erkennbaren) Kanten:



- Da die isotrope homogene Diffusion also der Gauß-Filterung entspricht, stellt sie keine Neuerung für die Bildverbesserung dar.

# Isotrope inhomogene Diffusion (1)

---

Die **isotrope inhomogene Diffusion** (Variante 2) variiert den Diffusionstensor lokal

- ~ Damit kann ein unterschiedliches Verhalten für Kanten und für abgebildete Objektflächen angesteuert werden
- ~ Im Idealfall
  - erwirkt die Diffusion den Ausgleich von Störungen auf abgebildeten homogenen Objektflächen
  - während an Kanten keine Diffusion statt findet

## Isotrope inhomogene Diffusion (2)

**Aber:** der Verlauf von Kanten ist i.A. nicht a priori bekannt

~ Daher sind Kanten für jede Zeit  $t$  zu schätzen – und zwar durch den Gradienten

$$\nabla u(x,y,t) = (\partial u / \partial x, \partial u / \partial y)$$

~ je größer der Gradientenbetrag ist, desto geringer sollte die Diffusion sein

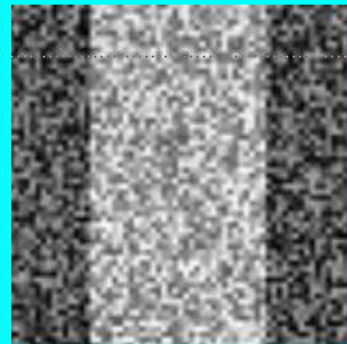
~ dazu muss der Diffusionstensor den Gradientenbetrag berücksichtigen:

$$\mathbf{D} = \begin{pmatrix} \varepsilon(\|\nabla u\|^2) & 0 \\ 0 & \varepsilon(\|\nabla u\|^2) \end{pmatrix} \text{ mit } \varepsilon(\|\nabla u\|^2) = \varepsilon_0 \frac{\lambda^2}{\|\nabla u\|^2 + \lambda^2}. \quad (7)$$

- Der Parameter  $\lambda$  steuert die Abnahme der Diffusion bei steigendem Gradienten:
  - großes  $\lambda$  : die inhomogene Diffusion nähert sich der homogenen Diffusion an
  - kleines  $\lambda$  : Diffusion sinkt rasch in Gebieten mit hohen Gradienten

# Isotrope inhomogene Diffusion (3)

- Die isotrope inhomogene Diffusion ist nicht mehr analytisch lösbar, sondern erfordert die genannte iterative Variante (s. Gleichung 5).
- Für die Anwendung der isotropen inhomogenen Diffusion auf ein Bild mit Kanten sollte bei gut gewähltem  $\lambda$  gelten:
  - der Intensitätsausgleich und damit die Rauschunterdrückung erfolgt nur in den die Objektflächen abbildenden Bildregionen
  - hinreichend stark ausgeprägte Kanten bleiben dabei erhalten



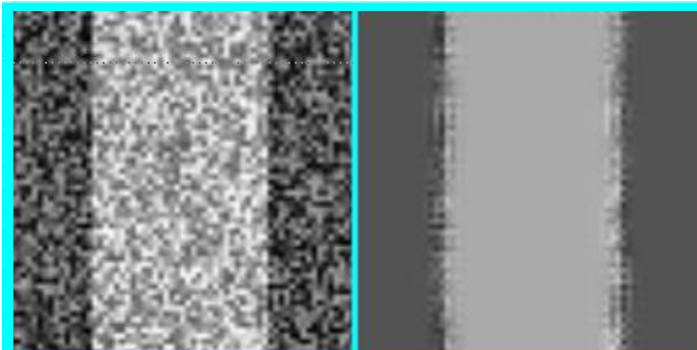
Isotrope inhomogene Diffusionsfilterung eines verrauschten Bildes nach 500 Iterationen.

Bildquelle: Klaus Tönnies: Grundlagen der Bildverarbeitung, Pearson Studium, 2005.

# Isotrope inhomogene Diffusion (4)

---

- Wie das Medianfilter führt die isotrope inhomogene Diffusion also zu einer Rauschunterdrückung auf abgebildeten homogenen Objektflächen und zu einer Erhaltung der abgebildeten Objektkanten.
- Dafür muss  $\lambda$  hinreichend klein zu wählen, um die Diffusion an den Kanten zu reduzieren!
- Damit wird das Glätten der Kanten verhindert, aber auch das Rauschen nahe der Kanten nicht hinreichend unterdrückt.



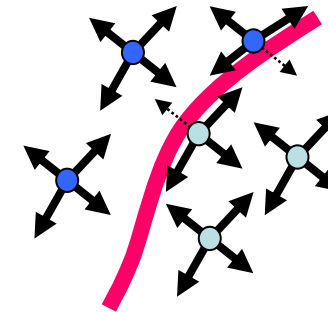
Isotrope inhomogene Diffusionsfilterung eines verrauschten Bildes nach 500 Iterationen.

Bildquelle: Klaus Tönnies: Grundlagen der Bildverarbeitung, Pearson Studium, 2005.

# Anisotrope inhomogene Diffusion (1)

---

- Bei der isotropen inhomogenen Filterung verblieb das Rauschen an den Kanten, weil die **richtungsunabhängige** Diffusion in Abhängigkeit der Gradienten an den Kanten gedämpft wurde.
- Dies Rauschen an den Kanten kann aber unterdrückt werden, wenn die Diffusion in Abhängigkeit vom Gradienten **nur in bestimmte Richtungen** zugelassen wird – nämlich bei Kanten nur **parallel zur Kantenrichtung!**
- Der dazu nötige Diffusionstensor wird aus einer Eigenwertzerlegung definiert. Diffusionstensoren sind positiv-definite, symmetrische Matrizen und haben daher zwei verschiedene Eigenwerte  $\lambda_1$  und  $\lambda_2$ .



Bildquelle: Klaus Tönnies:  
Grundlagen der Bildverarbeitung, Pearson Studium,  
2005.



## Anisotrope inhomogene Diffusion (2)

---

- Eigenwertzerlegung des Diffusionstensors:

$$\mathbf{D} = \begin{pmatrix} e_{1,1} & e_{2,1} \\ e_{1,2} & e_{2,2} \end{pmatrix} \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} \begin{pmatrix} e_{1,1} & e_{1,2} \\ e_{2,1} & e_{2,2} \end{pmatrix} \quad (8)$$

mit Eigenwerten  $\lambda_1$  und  $\lambda_2$  und Eigenvektoren  $(e_{1,1}, e_{1,2})^T$  bzw.  $(e_{2,1}, e_{2,2})^T$

- ~ Die Eigenwerte  $\lambda_1$  und  $\lambda_2$  spezifizieren die Stärke der Diffusion
- ~ Zur Blockierung der Diffusion über Kanten muss einer der beiden Eigenvektoren in Gradientenrichtung zeigen und einen niedrigen Eigenwert haben
- ~ Der zweite Eigenvektor, der orthogonal zum ersten Eigenvektor ist, verläuft parallel zur Kante. Seine Diffusionsstärke soll unabhängig von der Kantenstärke sein und zeigt entsprechend einen konstanten Eigenwert

# Anisotrope inhomogene Diffusion (3)

- Die folg. Eigenvektoren und -werte erfüllen die genannten Bedingungen:

$$\begin{pmatrix} e_{1,1} & e_{1,2} \end{pmatrix} = \frac{\nabla u}{\|\nabla u\|}, \quad \lambda_1 = \varepsilon(\|\nabla u\|^2) = \varepsilon_0 \frac{\lambda^2}{\|\nabla u\|^2 + \lambda^2}, \quad (9)$$

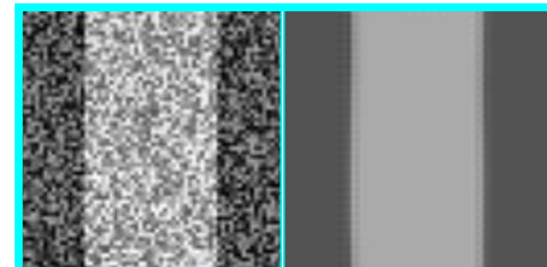
$$\begin{pmatrix} e_{2,1} & e_{2,2} \end{pmatrix} = \begin{pmatrix} e_{1,2} & -e_{1,1} \end{pmatrix}, \quad \lambda_2 = 1.$$

$\lambda_1, \lambda_2$  Eigenwerte  
 $\lambda$  = Steuerparameter

- Erster Eigenvektor in Richtung des Gradienten und Eigenwert sinkt mit Gradientenstärke: keine Diffusion über Kanten hinweg.
- Zweiter Eigenvektor parallel zur Kante und Eigenwert konstant: Diffusion parallel zur Kante ist von der Kantenstärke unabhängig.

# Anisotrope inhomogene Diffusion (4)

- Die anisotrope Diffusion ist wie die isotrope inhomogene Diffusion iterativ zu lösen (s. Gleichung 5).
- Die anisotrope Diffusion findet auch an Kantenpositionen  $(x,y)$  statt, wenn parallel zur Kantenrichtung Intensitätsunterschiede (dann  $\nabla u(x,y) \neq 0$ ) vorliegen.
- Intensitätsunterschiede orthogonal zur Kantenrichtung gehen durch  $\varepsilon(\|\nabla u\|^2)$  kaum ins Ergebnis ein.



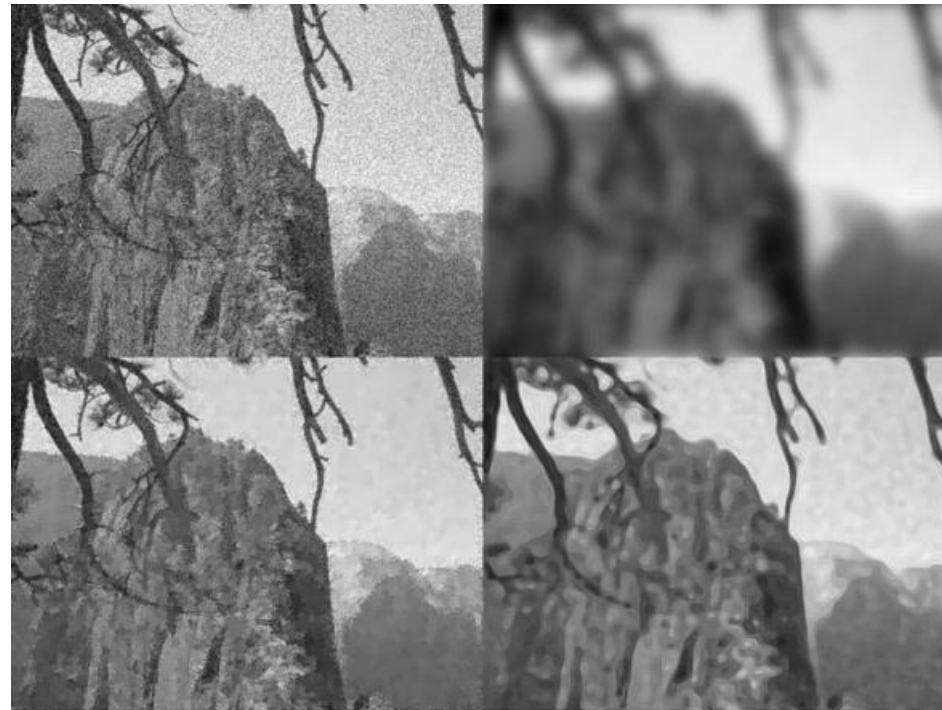
Anisotrope Diffusionsfilterung nach 500 Iterationen.

Bildquelle: Klaus Tönnies: Grundlagen der Bildverarbeitung, Pearson Studium, 2005.

# Anisotrope inhomogene Diffusion (5)

---

Verrauschtes Bild (oben links), homogene Diffusion (oben rechts),  
isotrope inhomogene Diffusion (unten links) und anisotrope  
Diffusion (unten rechts) nach 20 Iterationen:



## Anisotrope inhomogene Diffusion (6)

---

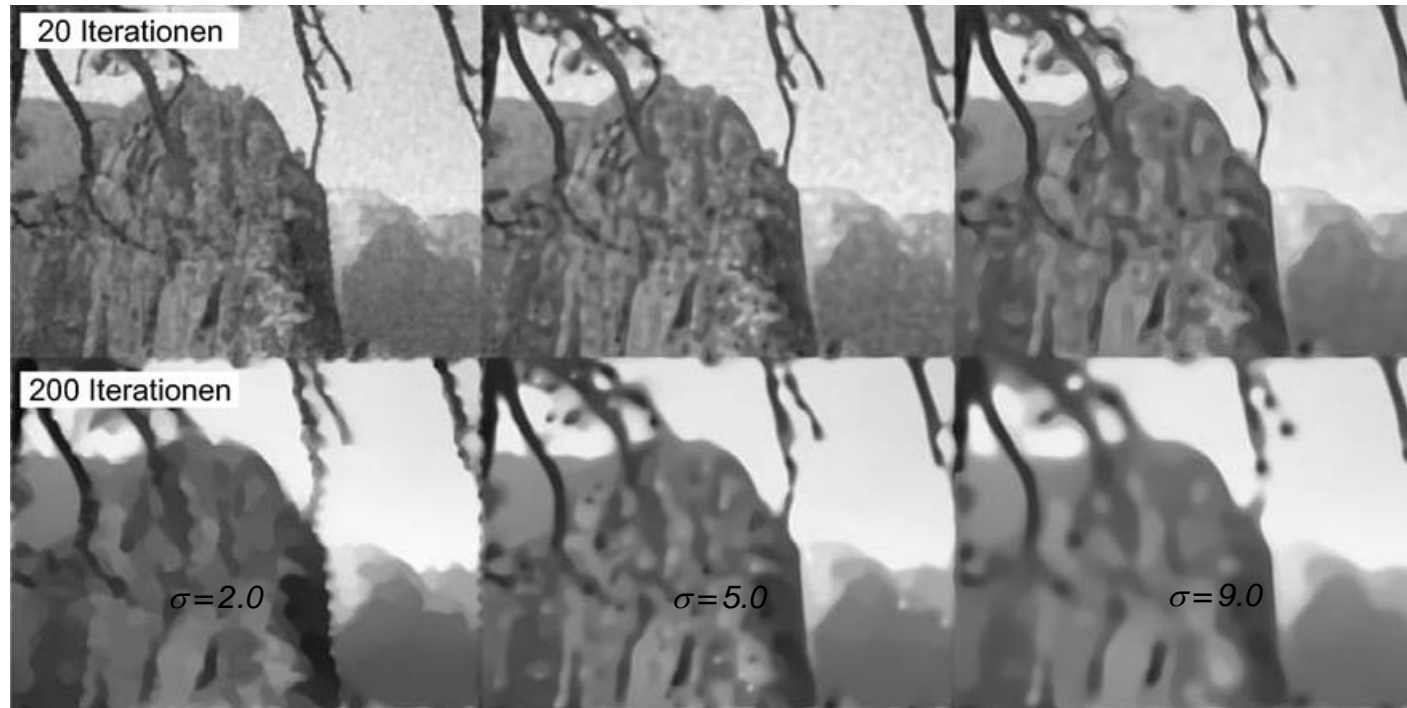
- Auch die anisotrope Diffusion wird nach vielen Iteration zu einer Kantenglättung, da der Gewichtungsterm  $\varepsilon(\|\nabla u\|^2)$  nie vollständig gleich Null sein wird.
- Dem kann durch eine **Entkopplung der Modelle von Rauschen und Kanten** begegnet werden:
  - der Konzentrationsgradient  $\nabla u$  wird durch einfache Differenzen abgeschätzt,
  - Im Diffusionstensor **D** wird anstelle von  $\nabla u$  in Gleichung (9) ein Kantengradient  $\nabla k$  eingeführt, der durch Kombination eines Differenzfilters mit einem glättenden Gauß-Filter angenähert wird.
- Durch entsprechende Wahl vom  $\sigma$  des Gauß-Filters werden alle Kanten, die kleiner als die Filtergröße sind, als Rauschen betrachtet und durch Diffusion aufgelöst.
- Das  $\sigma$  des Gauß-Filters trennt also zwischen feinem Rauschen in hohem Frequenzbereich und deutlichen Kanten in niedrigerem Frequenzbereich.

# Anisotrope Diffusion (6)

---

Anisotrope Diffusion für unterschiedliche Filtergrößen  $\sigma$  zur Bestimmung der Kantengradienten  $k$ .

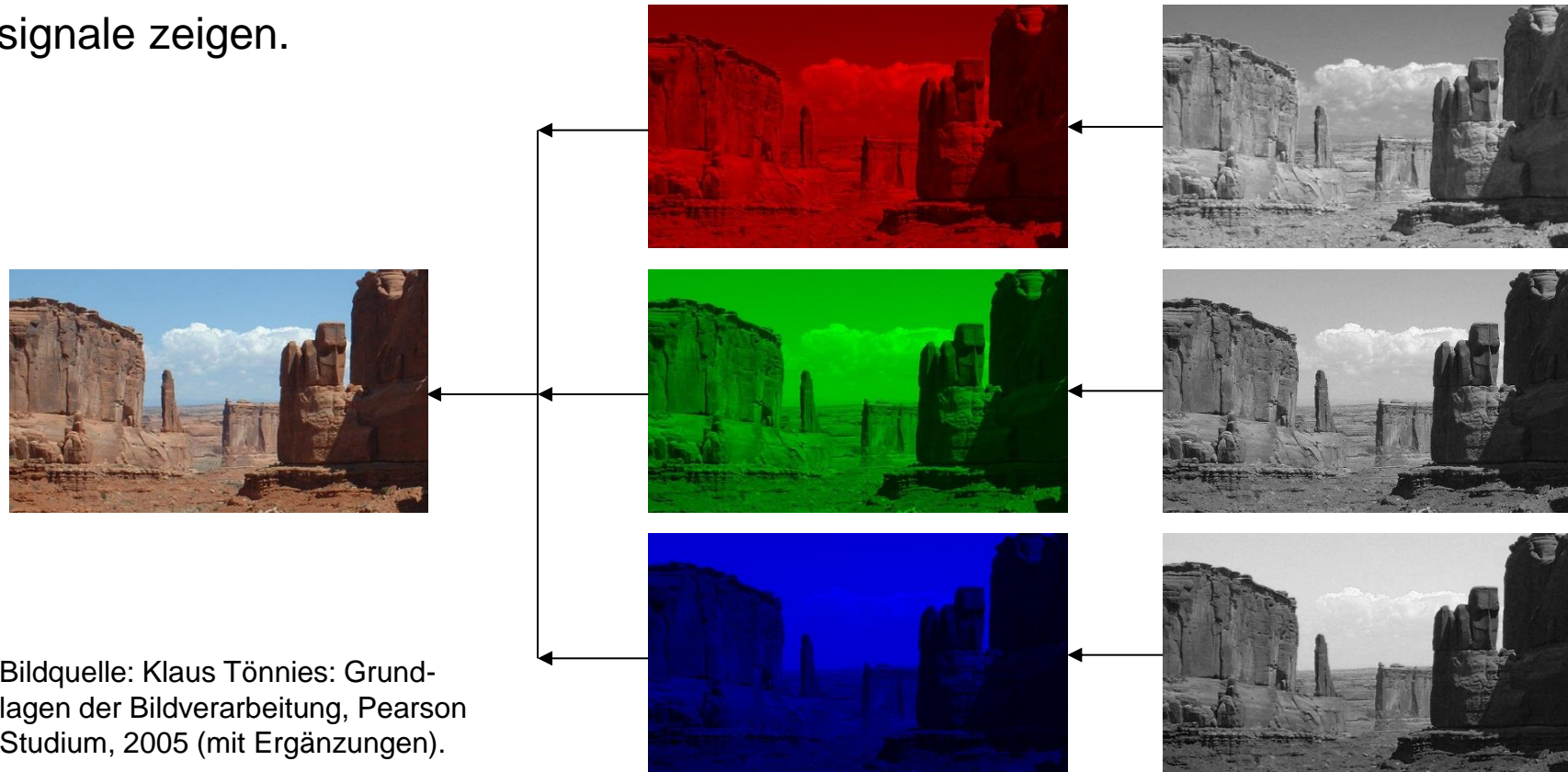
Die Filterung bewirkt, dass Details, die kleiner als die Filtergröße  $\sigma$  sind, als Rauschen betrachtet und durch Diffusion aufgelöst werden.



# Filterung von Farbbildern (1)

Die verschiedenen Kanäle von Mehrkanalbildern zeigen i.A. verschiedene Intensitätswerte.

Für RGB-Farbbilder können also  $I_R(x,y)$ ,  $I_G(x,y)$  und  $I_B(x,y)$  grundsätzlich unterschiedliche Intensitätswerte und damit auch unterschiedliche Stör- und Kanten-signale zeigen.



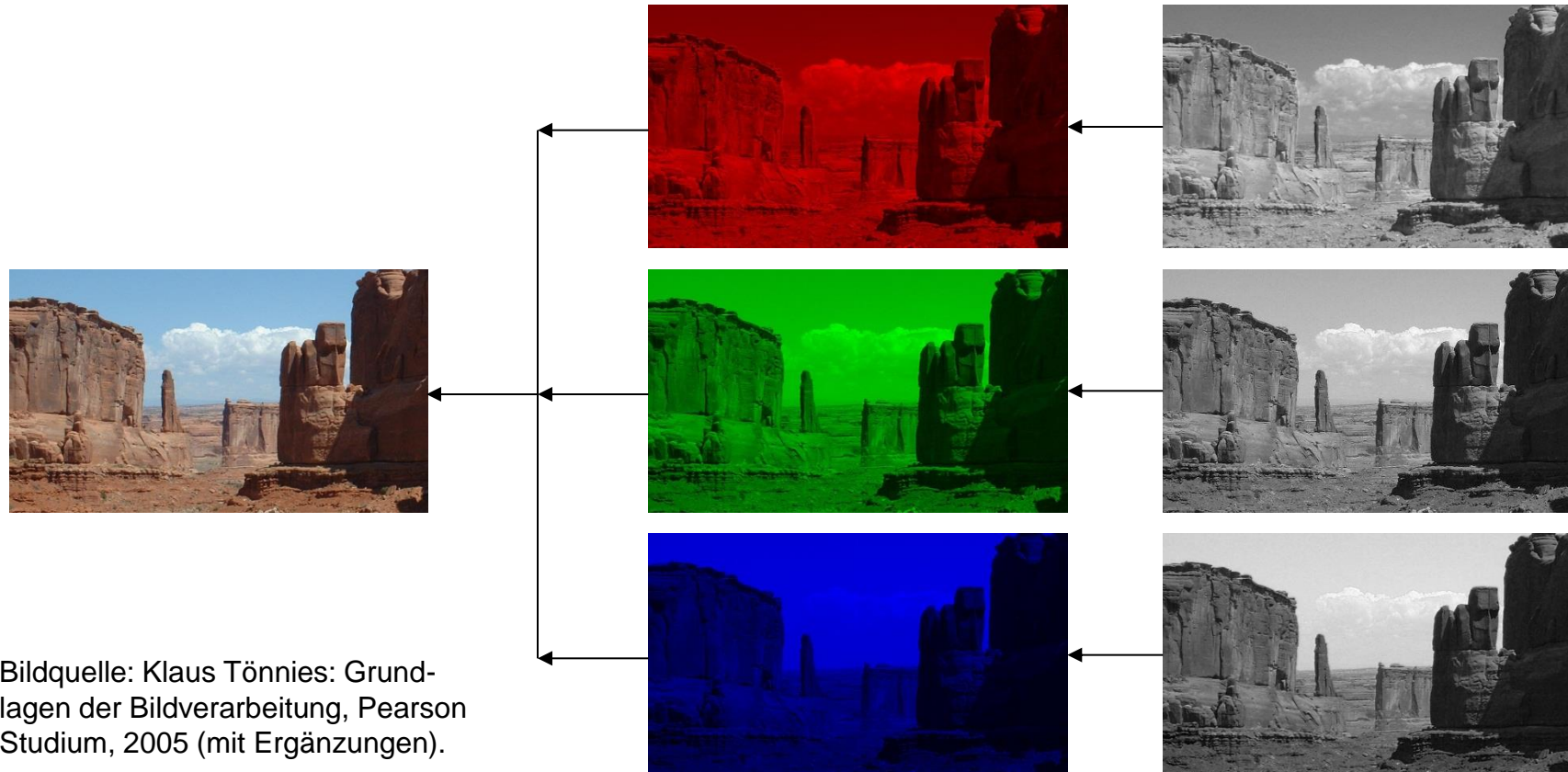
Bildquelle: Klaus Tönnies: Grundlagen der Bildverarbeitung, Pearson Studium, 2005 (mit Ergänzungen).



## Filterung von Farbbildern (2)

In der Abbildung sind z.B. die Konturen der Wolken gegenüber dem Himmel im Blaukanal deutlich schwächer ausgeprägt.

Damit stellt sich die Frage, in welcher Art Glättungs- und Kantenfilter auf Mehrkanalbildern, hier für RGB-Bilder, anzuwenden sind.

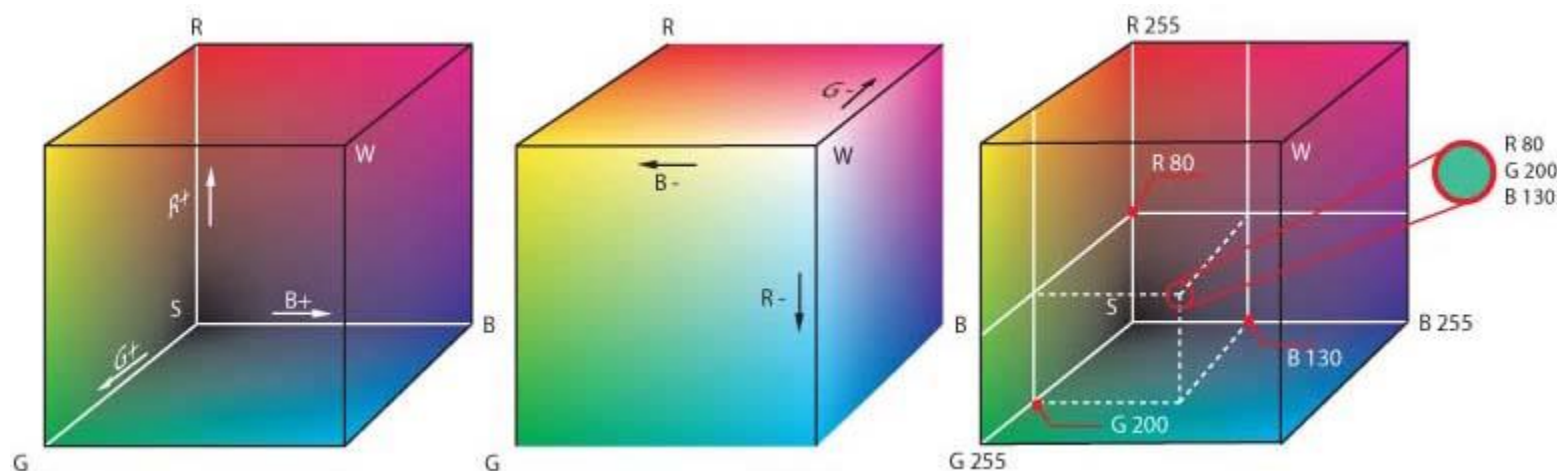


Bildquelle: Klaus Tönnies: Grundlagen der Bildverarbeitung, Pearson Studium, 2005 (mit Ergänzungen).



# Das RGB-Farbmodell

- Grundsätzlich gibt es verschiedene Farbmodelle, um die in Farbbildern enthaltene Information zu kodieren.
- Das RGB-Farbmodell ist ein additives Farbmodell, das Farbwahrnehmungen durch das additive Mischen der drei Grundfarben *Rot*, *Grün*, *Blau* nachbildet.
- Sind ein Rot, ein Grün und ein Blau in maximaler Intensität definiert, so kann eine bestimmte Farbe F durch ihren Rotanteil R, ihren Grünanteil G und ihren Blauanteil B beschrieben werden:  $I_{RGB}(x,y) = (I_R(x,y), I_G(x,y), I_B(x,y))$ .



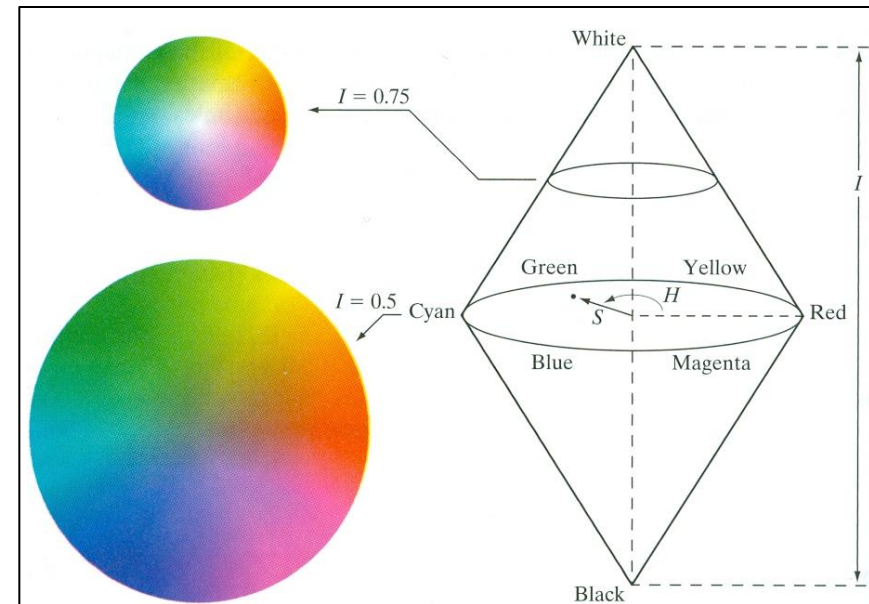
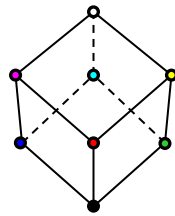
# HSI-Farbmodell (1)

Das HSI-Farbmodell wird im Computersehen sowie in der Farbphotographie und Videotechnik verwendet und repräsentiert Farbbilder durch drei Größen:

- **H** für **hue** (Färbung, Tönung)  $\in [0, 2\pi)$  \*
- **S** für **saturation** (Sättigung, Chroma)  $= 1 - 3 \cdot \min\{R, G, B\} / (R + G + B)$ ;  $S = 0$  für Schwarz
- **I** für **intensity** (Intensität, Helligkeit)  $= (R + G + B) / 3$


Das HSI-Farbmodell entkoppelt also die Intensitätsinformation von der Farbinformation (Färbung und Sättigung)!

Ferner entspricht das HSI-Farbmodell besser der Farbempfindung

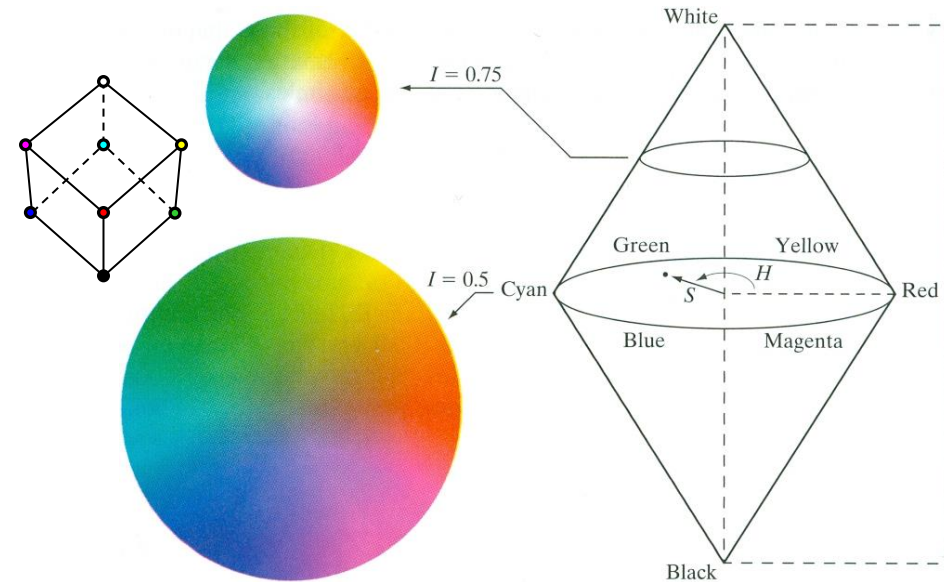


\* s. Anhang für Konvertierungen

## HSI-Farbmodell (2)

Color	R	G	B	H	s	i
	0	0	0	n.a.	0	0
	255	0	0	0	1	0.33
	0	255	0	120	1	0.33
	0	0	255	240	1	0.33
	255	255	0	60	1	0.66
	255	255	255	n.a.	0	1

R, G, B in [0, 255], H in [0, 360], s, i in [0, 1]



Bildquelle: R. C. Gonzales, R. E. Woods: Digital Image Processing (3<sup>rd</sup> Ed.). Pearson Ed. Int. 2008

- Der HSI-Farbraum zeigt eine vertikale Intensitätsachse sowie die Längen (Sättigungen) und Winkel (Farbtöne) zur Rotachse der Vektoren zu den Farbpunkten.
- Die Grundfarben Rot, Grün und Blau zeigen Winkelabstände von  $120^\circ$  und Intensitätswerte von 0,33.
- Die HSI-Wertetripel sind also nicht beliebig innerhalb ihrer Wertebereiche belegbar. Dies zeigt der Wertebereich der graphischen Farbraumdarstellung sowie die Herleitungen aus den RGB-Kodierungen.

# Glättung von Farbbildern (1)

---

Die Glättung von Einkanalbildern wurde umgesetzt,

- indem ein Glättungsfilter (Mittelwertfilter, Gauß-Filter, Binomialfilter oder Medianfilter) auf jedes Pixel angewendet wird:

Dieser Ansatz ist auf RGB-Farbbilder übertragbar:

- indem das Filter jetzt nur nicht mehr auf einen Intensitätswert  $I(x,y)$ , sondern auf ein Intensitätstripel angewendet wird:

$$I_{RGB}(x,y) = [ I_R(x,y), I_G(x,y), I_B(x,y) ] .$$

## Glättung von Farbbildern (2)

---

Der Einfachheit halber betrachten wir das Mittelwertfilter:

Sei  $N(x,y)$  die durch die Filtermaske definierte Nachbarschaftsumgebung mit  $K$  Pixeln und zentriert im Pixel  $I_{RGB}(x,y)$  eines RGB-Farbbildes.

Der Mittelwertvektor der RGB-Intensitätsvektoren ist dann:

$$I_{RGB_{av}}(x,y) = 1/K \cdot \sum_{(s,t) \in N(x,y)} I(s,t).$$

Nach den Gesetzen der Vektoraddition:

$$I_{RGB_{av}}(x,y) = [ 1/K \cdot \sum_{(s,t) \in N(x,y)} I_R(s,t), 1/K \cdot \sum_{(s,t) \in N(x,y)} I_G(s,t), 1/K \cdot \sum_{(s,t) \in N(x,y)} I_B(s,t) ]$$

# Glättung von Farbbildern (3)

---

- Die folgende Folie zeigt
  - das RGB-Farbbild *Lena*
  - seine RGB-Komponenten
  - seine HSI-Komponenten
- Jeder RGB-Kanal wurde separat mit einem  $5 \times 5$ -Mittelwertfilter geglättet
  - ↪ Die so geglätteten Farbkanäle werden zu einem resultierenden geglätteten RGB-Farbbild kombiniert
- Alternativ die HSI-Repräsentation ausschließlich im Intensitätskanal geglättet (die Tönungs- und Sättigungskanäle bleiben unverändert)
  - ↪ Das Glättungsergebnis auf dem HSI-Intensitätskanal wird wieder in die RGB-Repräsentation konvertiert



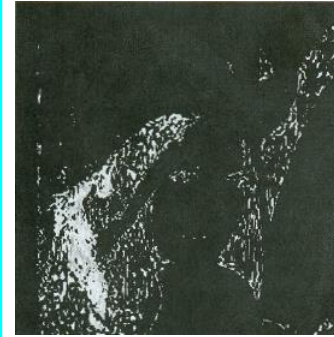
# Glättung von Farbbildern (4)



RGB *Lena*



red component



hue component



RGB smoothing



green component



saturation



blue component



intensity



HSI smoothing

Bildquelle: R. C. Gonzalez,  
R. E. Woods: Digital Image  
Processing (3<sup>rd</sup> Ed.). Pear-  
son Education Intern., pp.  
441ff., 2008.

# Glättung von Farbbildern (5)

---

Beide Ergebnisse der Mittelwertglättung sehen zunächst ähnlich aus.

Das Differenzbild zeigt, dass beide Ergebnisse nicht gleich sind:

- Die RGB-Glättung konstituiert sich aus den *Mittelungen über den Farbwerten* der Pixel. Also ergibt sich i.A. eine veränderte Farbe für jeden Pixel.
- Bei der HSI-Glättung wird lediglich die Intensität gemittelt, nicht aber die Tönung und die Saturation. Die Farbe wird also nicht geändert!

Diese Unterschiede nehmen mit wachsender Filtergröße zu.



RGB smoothing



difference



HSI smoothing



# Kantenhervorhebung in Farbbildern (1)

---

Analog zur Betrachtung der Glättung von Farbbildern wird die Erweiterung der Kantenhervorhebung auf Farbbildern betrachtet.

Der Einfachheit halber wird hier der  $L_4$ -Laplace-Filter betrachtet:

$$L_4 = \begin{pmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{pmatrix}.$$

Aus den Gesetzen der Vektoraddition folgt:

$$\nabla^2 I_{RGB}(x,y) = [ \nabla^2 I_R(x,y) , \nabla^2 I_G(x,y) , \nabla^2 I_B(x,y) ]$$

## Kantenhervorhebung in Farbbildern (2)

---

Wieder wird eine Laplace-Kantenhervorhebung auf den RGB-Kanälen mit einer Laplace-Kantenhervorhebung auf dem Intensitätskanal verglichen.

Wieder zeigt das Differenzbild Unterschiede.

Wieder erklärt das Mischen der Farbinformation bei der RGB-Kantenhervorhebung die Unterschiede zur HSI-Kantenhervorhebung, die ausschließlich auf dem Intensitätskanal ohne Manipulation der Farben durchgeführt wird.



RGB sharpening



difference



HSI sharpening

# Fazit zur Filterung auf Farbbildern

---

- Sowohl für Glättung als auch für Kantenhervorhebung auf Farbbildern ist das HSI-Farbmodell geeigneter.
  - Beide Arten der Filterung können dort separat auf dem Intensitätskanal, aus dem sich die Gradienten für Störungen und Kanten ergeben, angewendet werden.
  - Dadurch wird die Farbinformation nicht manipuliert.
- In der Praxis wird auch oft auf Grauwertvarianten von Farbbildern gearbeitet. Diese entsprechen dem Intensitätskanal. Natürlich muss dann diese Vernachlässigung der Farbinformation mit der Aufgabenstellung vereinbar sein.

# Zusammenfassung (1)

---

- **Nichtlineare Filter** können für schwierige Anwendungsfälle eine im Vergleich zu linearen Filtern effizientere Trennung zwischen Rauschunterdrückung und Kanten hervorhebung umsetzen.
- Zwei nichtlineare Filterklassen sind **Rangordnungsfiler** und **Diffusionsfilter**.
- **Rangordnungsfiler** sortieren die Intensitäten eines Pixels und der Pixel seiner lokalen Nachbarschaft und liefern den Intensitätswert, der an einem bestimmten Rang der sortierten Reihenfolge steht – z.B., erster, d. h. minimaler Wert, letzter, d.h. maximaler Wert, Median, ...
- Das **Medianfilter** liefert den Medianwert, unterdrückt effektiv das Impulsrauschen und ist bei lokal geradlinigem Kantenmodell kantenerhaltend.

## Zusammenfassung (2)

---

- **Diffusionsfilter** basieren auf dem Modell der Diffusion. Dabei wird die Intensitätsfunktion eines Bildes als räumliche Konzentrationsfunktion interpretiert.
- Nach der **isotropen homogenen Diffusion** sind nach großer Iterationszahl leicht alle Kanten vollständig verschwunden. Sie entspricht einer Gauß-Filterung.
- Bei der **isotropen inhomogenen** Diffusion bleibt der Unterschied zwischen Vordergrund und Hintergrund durch die Kantenbegrenzungen erhalten, allerdings bleiben die Kanten verrauscht.
- Die **anisotrope Diffusion** erfolgt nur parallel zur Kantenrichtung und nicht über die Kante hinaus. Daher bleibt der Unterschied zwischen Vordergrund und Hintergrund erhalten und das Rauschen an den Kanten wird unterdrückt.
- Die Kantenerhaltung der anisotropen Diffusion ist durch **Entkopplung der Modelle von Rauschen und Kanten** verbesserbar, indem zwischen störungsbedingtem Konzentrationsgradient  $\nabla u$  und Kantengradient  $\nabla k$  unterschieden wird.

## Zusammenfassung (3)

---

- Die Anwendung von Filtern zur Glättung und Kanten hervorhebung kann auf RGB-Farbbilder erweitert werden, indem dieselbe Filterung auf jedem Kanal separat angewendet wird.
  - Hierbei kommt es aber auch zur ungewollten Farbmanipulation.
- Die Anwendung von Filtern zur Glättung und Kanten hervorhebung auf HSI-kodierte Farbbilder zeigt diese Manipulation nicht, da die Intensitätsinformation, aus der sich Gradienten für Störungen und Kanten ergeben, von der Farb-information getrennt kodiert ist.

# Anhang: RGB zu HSI

## Converting colors from RGB to HSI

Given an image in RGB color format, the  $H$  component of each RGB pixel is obtained using the equation

$$H = \begin{cases} \theta & \text{if } B \leq G \\ 360 - \theta & \text{if } B > G \end{cases} \quad (6.2-2)$$

with

$$\theta = \cos^{-1} \left\{ \frac{\frac{1}{2}[(R - G) + (R - B)]}{[(R - G)^2 + (R - B)(G - B)]^{1/2}} \right\}.$$

The saturation component is given by

$$S = 1 - \frac{3}{(R + G + B)} [\min(R, G, B)]. \quad (6.2-3)$$

Finally, the intensity component is given by

$$I = \frac{1}{3} (R + G + B). \quad (6.2-4)$$

It is assumed that the RGB values have been normalized to the range  $[0, 1]$  and that angle  $\theta$  is measured with respect to the red axis of the HSI space, as indicated in Fig. 6.13. Hue can be normalized to the range  $[0, 1]$  by dividing by  $360^\circ$  all values resulting from Eq. (6.2-2). The other two HSI components already are in this range if the given RGB values are in the interval  $[0, 1]$ .

The results in Eqs. (6.2-2) through (6.2-4) can be derived from the geometry shown in Figs. 6.12 and 6.13. The derivation is tedious and would not add significantly to the present discussion. The interested reader can consult the book's references or web site for a proof of these equations, as well as for the following HSI to RGB conversion results.

# Anhang: HSI zu RGB (1)

---

## Converting colors from HSI to RGB

Given values of HSI in the interval  $[0, 1]$ , we now want to find the corresponding RGB values in the same range. The applicable equations depend on the values of  $H$ . There are three sectors of interest, corresponding to the  $120^\circ$  intervals in the separation of primaries (see Fig. 6.13). We begin by multiplying  $H$  by  $360^\circ$ , which returns the hue to its original range of  $[0^\circ, 360^\circ]$ .

*RG sector* ( $0^\circ \leq H < 120^\circ$ ): When  $H$  is in this sector, the RGB components are given by the equations

$$B = I(1 - S) \quad (6.2-5)$$

$$R = I \left[ 1 + \frac{S \cos H}{\cos(60^\circ - H)} \right] \quad (6.2-6)$$

and

$$G = 3I - (R + B), \quad (6.2-7)$$



## Anhang: HSI zu RGB (2)

---

*GB sector* ( $120^\circ \leq H < 240^\circ$ ): If the given value of  $H$  is in this sector, we first subtract  $120^\circ$  from it:

$$H = H - 120^\circ. \quad (6.2-8)$$

Then the RGB components are

$$R = I(1 - S) \quad (6.2-9)$$

$$G = I \left[ 1 + \frac{S \cos H}{\cos(60^\circ - H)} \right] \quad (6.2-10)$$

and

$$B = 3I - (R + G). \quad (6.2-11)$$

*BR sector* ( $240^\circ \leq H \leq 360^\circ$ ): Finally, if  $H$  is in this range, we subtract  $240^\circ$  from it:

$$H = H - 240^\circ. \quad (6.2-12)$$

Then the RGB components are

$$G = I(1 - S) \quad (6.2-13)$$

$$B = I \left[ 1 + \frac{S \cos H}{\cos(60^\circ - H)} \right] \quad (6.2-14)$$

and

$$R = 3I - (G + B). \quad (6.2-15)$$