

Bildrekonstruktion & Multiresolution

Verkleinern von Bildern



Was ist zu beachten?

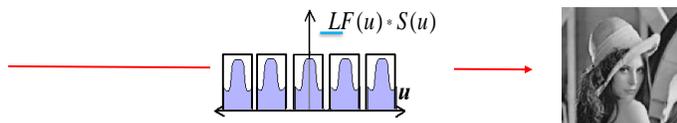
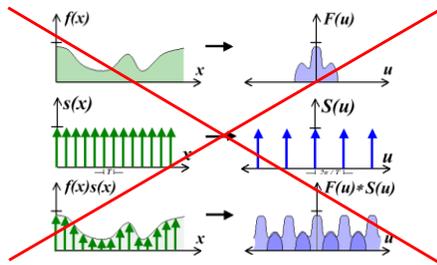
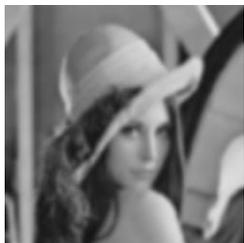
Es kann aliasing auftreten!

Das Abtasttheorem sagt wie man es vermeidet? ==>

Verkleinern von Bildern (2)

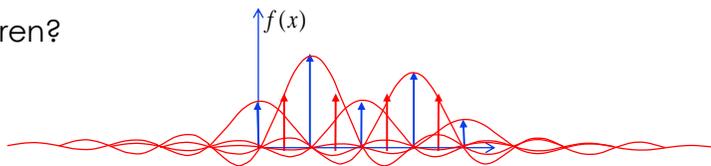


Vor dem Abtasten
Lowpass filtern!



Rekonstruktion abgetasteter Signale

Wie ist zu interpolieren?

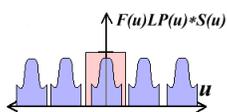


Gibt es den idealen Rekonstruktionsfilter? ?

Theoretisch ja --- praktisch nein?

Theorie:

falls das Signal korrekt Tiefpass gefiltert wurde, sollten alle verbliebenen Frequenzanteile voll rekonstruiert werden.
=> der Frequenzraum wird mit einem Rechteck multipliziert.



==> Faltung im Ortsraum.

$$f(x_i) = \sum f(n\Delta x)\delta(x-n\Delta x)$$

$$f(x) = \sum f(n\Delta x)\delta(x-n\Delta x) * \frac{1}{\Delta x} \frac{\sin(x/\Delta x)}{x/\Delta x}$$

Faltung mit unendlich ausgedehnten Funktionen nicht möglich!

Vergrößern von Bildern (Rekonstruktion)

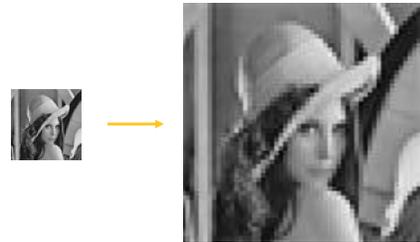
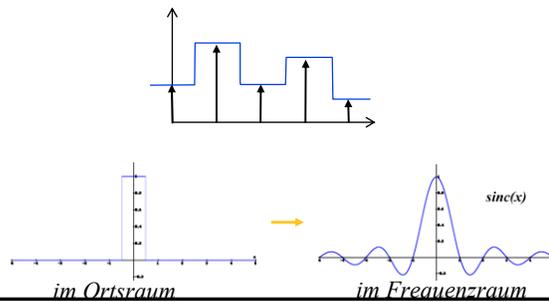
1. Interpolation durch Multiplikation mit Boxfilter im Frequenzraum!

⇔

2. Durch Faltung im Ortsraum!

3. Durch Interpolation im Ortsraum?

z.B. Nearest Neighbor

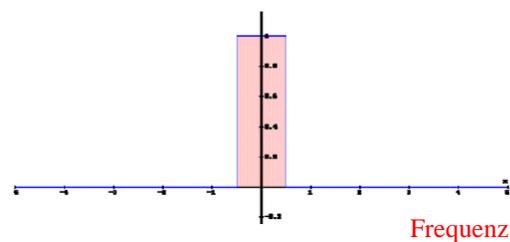
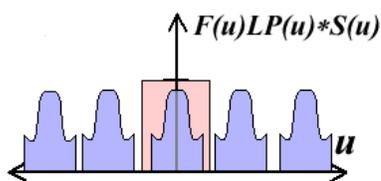
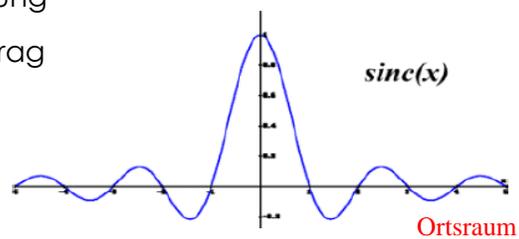


Der Ideale Rekonstruktions-Filter

Unendliche räumliche Ausdehnung

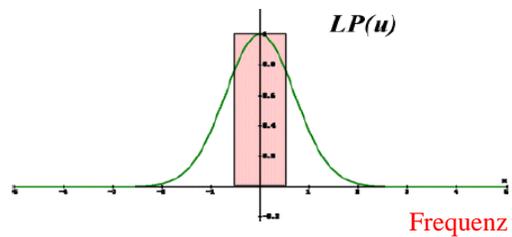
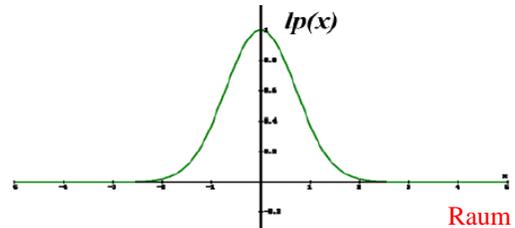
Jeder Messpunkt gibt einen Beitrag zu jedem interpolierten Punkt

teuer/unmöglich zu berechnen



Gauss Rekonstruktions-Filter

Bieten CRT-Bildschirme
ohne Zusatzaufwand!

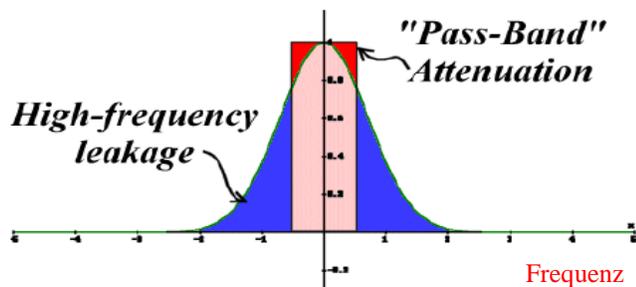


Probleme mit Rekonstruktions-Filter

Viele der sichtbaren Fehler in abgetasteten Bildern sind das Ergebnis schlechter Rekonstruktionsfilter.

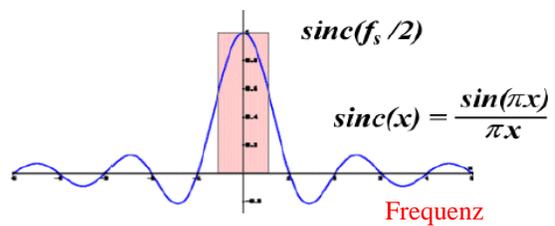
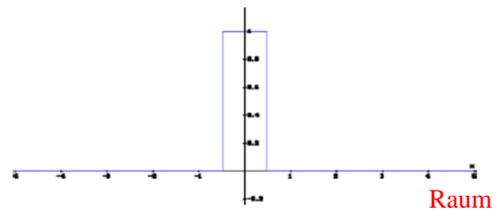
Abschwächung von Frequenzen resultiert in unscharfen Bildern

Eine Verstärkung hoher Frequenzen verursacht "ringing" oder eine Überbetonung des Abtastrasters.



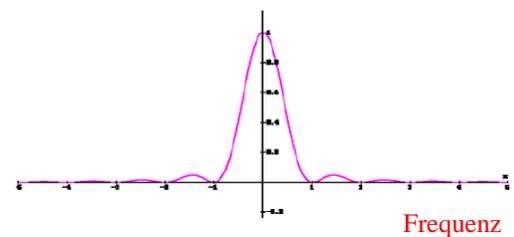
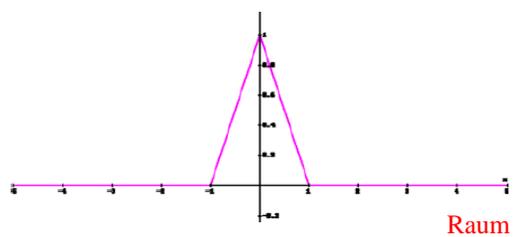
Box Filter / Nearest Neighbor

Macht Pixel
zu kleinen Quadraten.



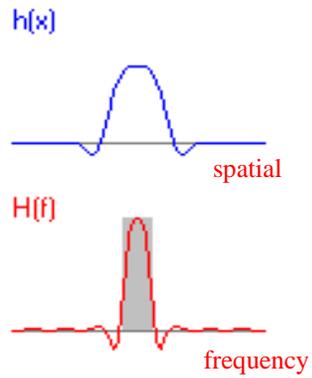
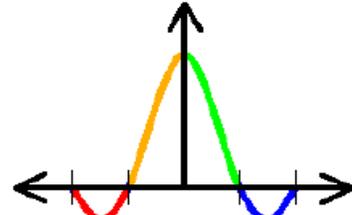
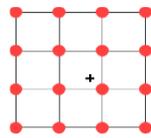
Tent Filter / Bi-Linear Interpolation

Einfach zu implementieren
relativ "glatt"



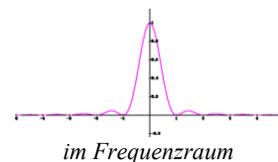
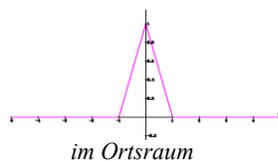
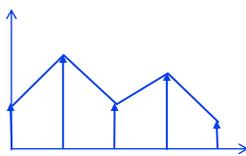
Bi-Cubic Interpolation

Einfache Approximation des
idealen Filters
(der sinc-Funktion)



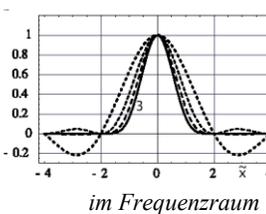
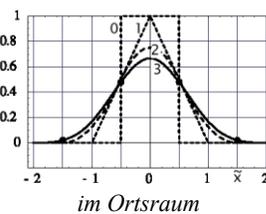
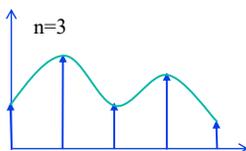
Interpolation

Lineare Interpolation



Spline Interpolation:

Interpolation mit Polynomen von Grad n C_0, C_1



Lokale Interpolationsregeln lassen sich immer als Faltung in Ortsraum
oder als Multiplikation im Frequenzraum verstehen.

Verkleinern von Bildern in der Praxis

Der ideale Tiefpassfilter benötigt eine Faltung mit einem unendlich ausgedehnten Kern.

In der Praxis werden lokale Faltungsoperatoren verwendet.

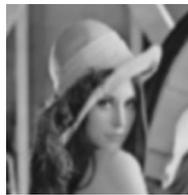
Als besonders günstig haben sich die Binomialmasken erwiesen, sie sind die diskrete Approximation einer Gaußfunktion.

$${}^4B = \frac{1}{256} \begin{pmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 6 \end{pmatrix} = \frac{1}{16} \begin{pmatrix} 1 \\ 4 \\ 6 \\ 4 \\ 1 \end{pmatrix} \cdot \frac{1}{16} (1 \ 4 \ 6 \ 4 \ 1)$$

Durch das Ausnutzen der Separabilität sowie der Symmetrie ergibt sich eine dramatisch Einsparung der Rechenschritte.



$G^{(0)}$



${}^4BG^{(0)}$



$G^{(1)} = R^4BG^{(0)}$

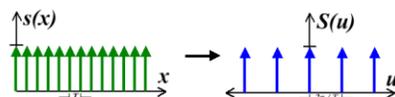
R : sei ein Reduktionsoperator der nur jeden 2ten Pixel abstastet.

Multi-Skalen-Darstellung

Bisher wurden alle Operationen entweder in Orts- oder Frequenzraum durchgeführt. Wurde in dem einen Raum eine lokale Untersuchung (Messung) durchgeführt, so konnte im anderen nur eine ganz globale Aussage getroffen werden (Unschärfe-Relation)!

Hohe Frequenzen benötigen eine hohe Auflösung!
und

Niedere Frequenzen benötigen eine niedere Auflösung!



Um Information im niederen (tiefen) Frequenzbereich zu bestimmen, muss mit großen Masken gefiltert werden!

→ großer Rechenaufwand.

Ein Ausweg: Die Gaußpyramide!

Gaußpyramide

Sei,

$R_{\downarrow 2}$: ein Glättungs- und Reduktionsoperator

$G^{(n)}$: ein Bild in der Auflösungsebene n

$$G^{(1)} = R_{\downarrow 2} G^{(0)}$$

$$G^{(2)} = R_{\downarrow 2} G^{(1)} = R_{\downarrow 2} (R_{\downarrow 2} G^{(0)})$$

.....

$$G^{(n)} = R_{\downarrow 2} G^{(n-1)} = \left(\prod_{i=1}^n R_{\downarrow 2} \right) G^{(0)}$$

Eine Menge $\{G^{(0)}, G^{(1)}, \dots, G^{(n)}\}$

heißt **Gaußpyramide** mit n Stufen oder Ebenen.

$G^{(0)}$



$G^{(1)}$



$G^{(2)}$



$G^{(3)}$



Von der Gauß-

zur

Laplacepyramide



\ominus

\equiv



\ominus

\equiv



\ominus

\equiv



\ominus

\equiv

$E^{\uparrow 2}$

$E^{\uparrow 2}$

$E^{\uparrow 2}$

Laplacepyramide

Sei,

$E^{\uparrow 2}$: ein Interpolations- und Expansionsoperator

B : ein Glättungsoperator

$G^{(n)}$: ein Bild in der Auflösungsebene n

Die Laplacepyramide kann als Bandpaßzerlegung eines Bildes verstanden werden. Aufgrund nicht perfekter Reduktions- und Expansionsfilter überlappen die einzelnen Bänder

$$L^{(0)} = G^{(0)} - E^{\uparrow 2} G^{(1)} = G^{(0)} - B G^{(0)} \\ = (I - B) G^{(0)}$$

.....

$$L^{(p)} = (I - B) G^{(p)}$$

Rekonstruktion des original Bildes
aus der Laplacepyramide

$$G^{(n-1)} = L^{(n-1)} + E^{\uparrow 2} G^{(n)}$$



Der Speicherplatzbedarf der
gesamten Pyramide eines Bildes
beträgt $4/3$ von dem des Bildes.

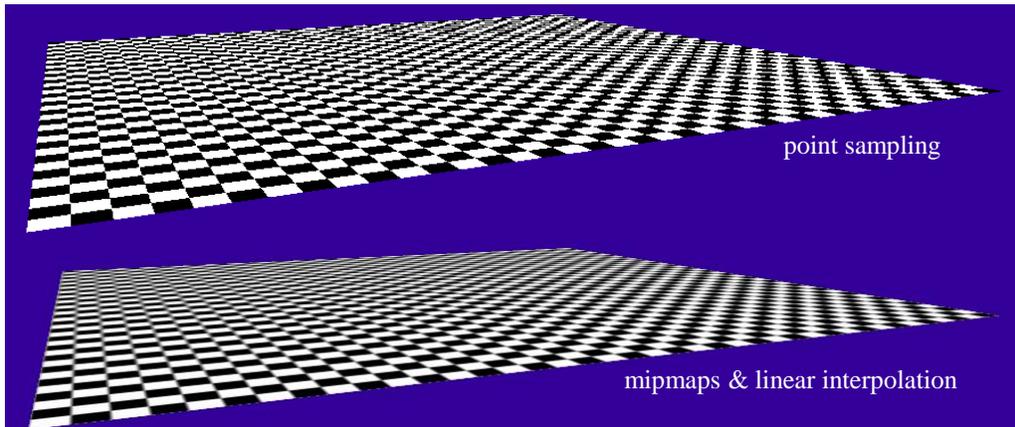
Anwendungen der Gauß- und Laplacepyramide

1. In der Computer Graphik
 - A) beim Überblenden oder Aneinanderhängen von Bildern.
 - B) Beim Abbilden von Texturen auf Oberflächen

2. Die Hauptanwendung dieser Auflösungs- und Laplacepyramide ist bei „grob-zu-fein“ Strategien im Bereich der Computational Science im Speziellen beim Maschinensehen und der Bildverarbeitung.
Ein kleiner Filter angewandt auf ein kleines Bild, entspricht einem großen Filter in einer höheren Auflösung. Der Rechenaufwand ist jedoch viel geringer!!

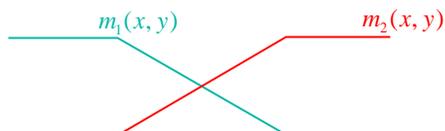
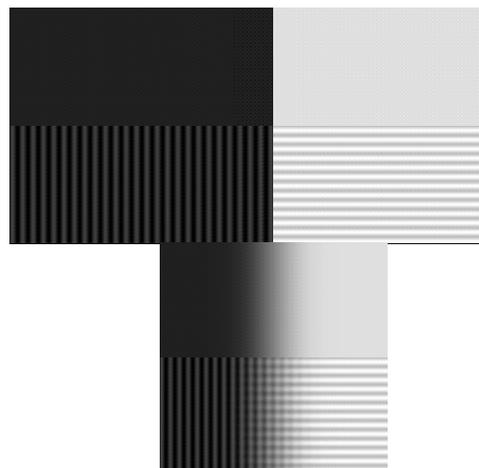
Aliasing in Texturen

Textur Fehler



Überblenden von Bildern

Aufgabe: Mehrere Bilder sollen zu einer Bildmosaik verbunden werden, und die Ränder sollen so unauffällig wie möglich sein.



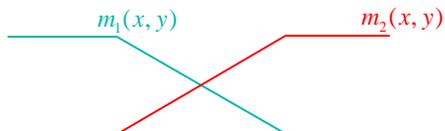
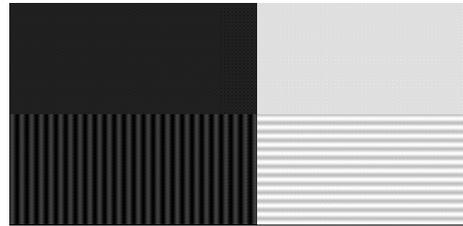
$$I_{neu}(x, y) = m_1(x, y)I_1(x, y) + m_2(x, y)I_2(x, y)$$

mit $m_1(x, y) + m_2(x, y) = 1$

Gibt es noch andere Lösungen?

Überblenden im Frequenzbereich

Idee: Niedere Ortsfrequenzen sollen eine längere Reichweite bekommen als hohe!



$$L^{(p)}_{neu}(x, y) = m^{(p)}_1(x, y)L^{(p)}_1(x, y) + m^{(p)}_2(x, y)L^{(p)}_2(x, y)$$

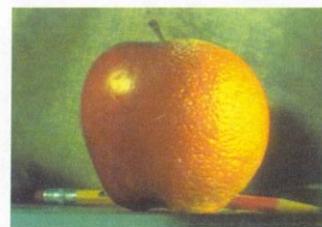
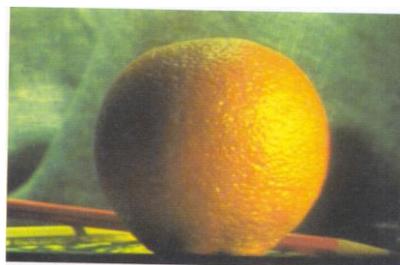
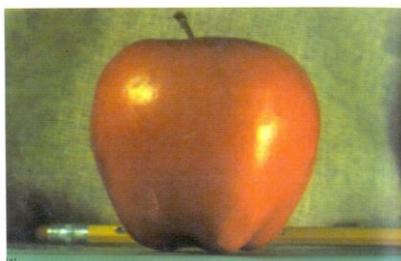
mit $m^{(p)}_i = G^{(p)}_i$ wobei $G^{(0)}_i = m_i$

und $k(m^{(p)}_1(x, y) + m^{(p)}_2(x, y)) = 1$

Burt, P.J.; Adelson, E.H.: The Laplacian pyramid as a compact image code. *IEEE Trans. Commun.* 31(1983) 802-540

Pyramid Blending

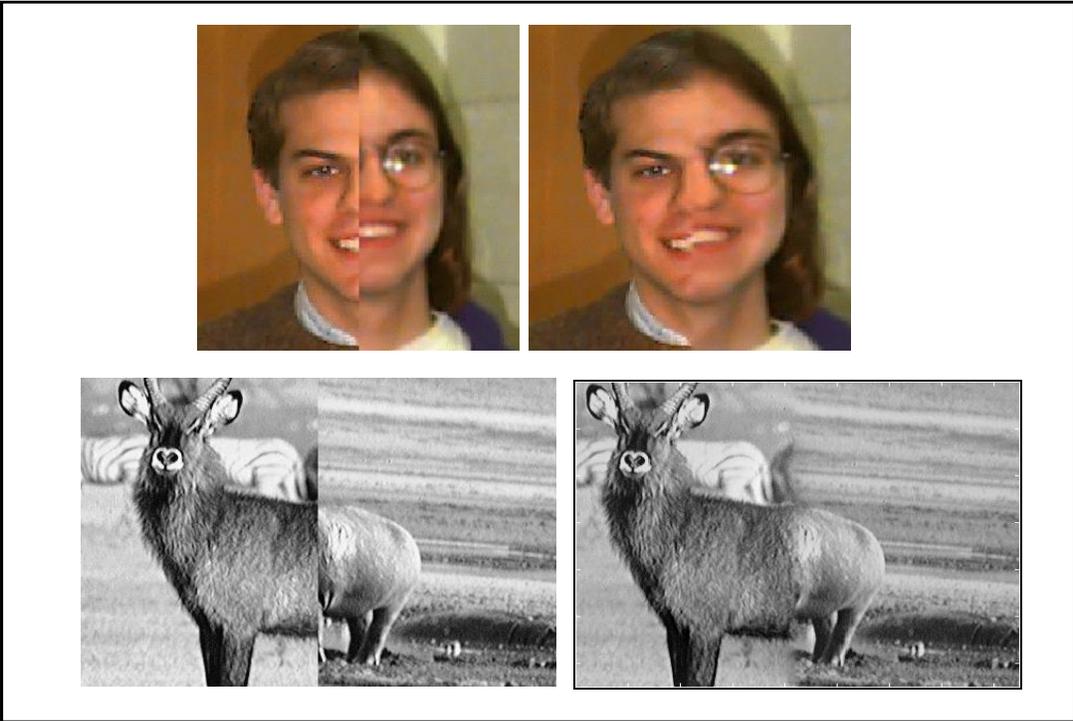
Burt, P.J.; Adelson, E.H.: The Laplacian pyramid as a compact image code. *IEEE Trans. Commun.* 31(1983) 802-540



(d)

(h)

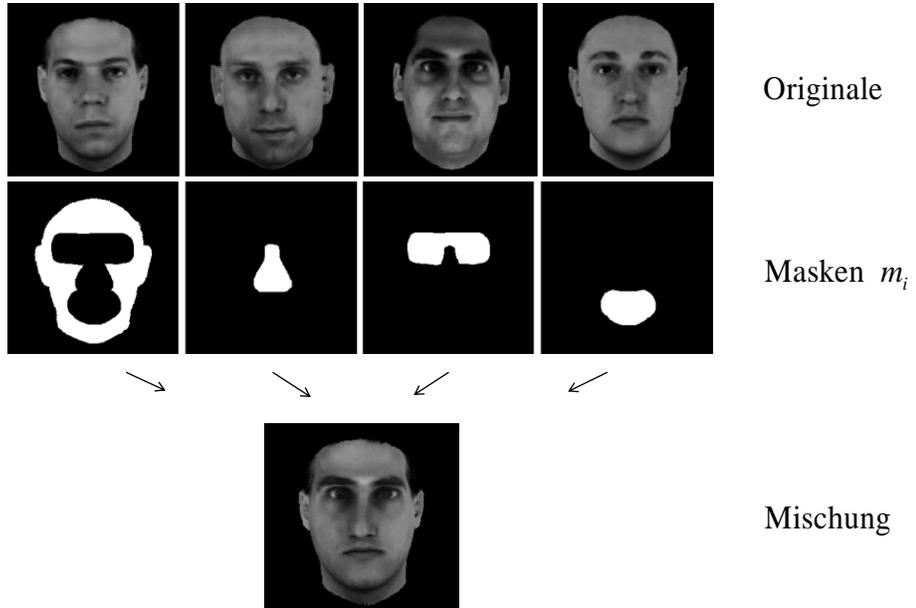
(l)



Regionales Überblenden



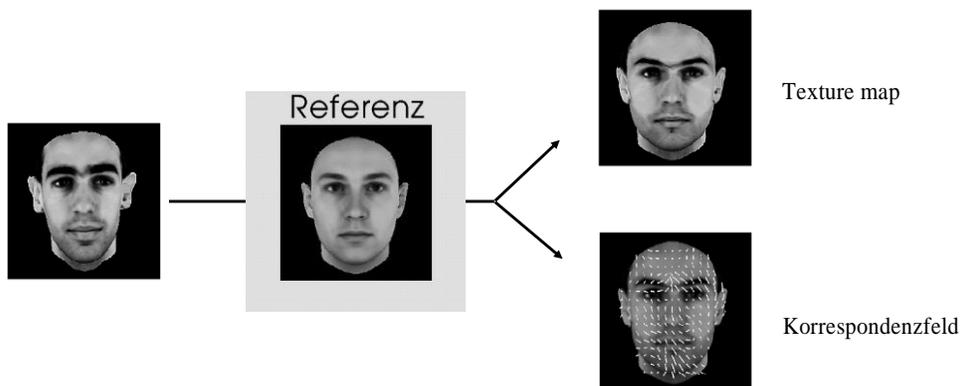
Ziel: Mischen von Gesichtern



Größenanpassung

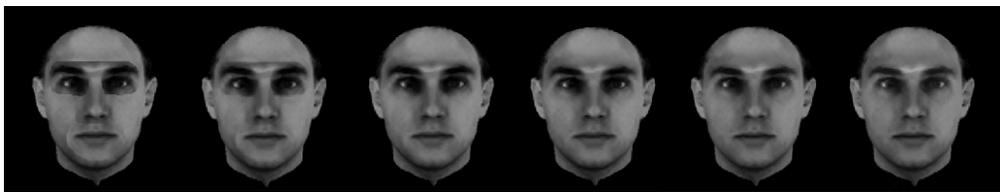
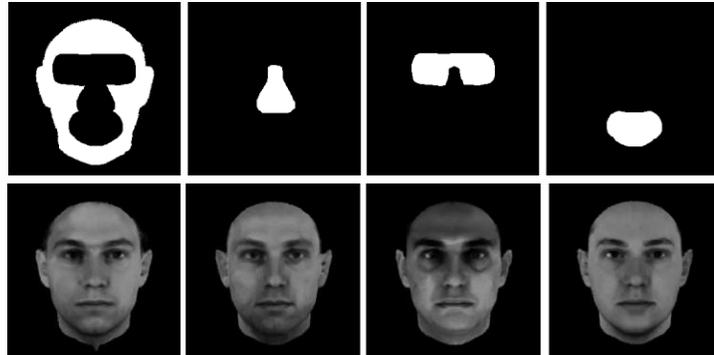
Problem: Gesichter sind unterschiedlich groß!

Lösung: Gesichter werden mittels Korrespondenz normiert.



Nun können Form und Textur separat überblendet werden →

Texture map blending



$level^{(0)}$

$l^{(1)}$

$l^{(2)}$

$l^{(3)}$

$l^{(4)}$

$l^{(5)}$

Überblenden in Form und Textur

$level^{(0)}$

$l^{(1)}$

$l^{(2)}$



$l^{(3)}$

$l^{(4)}$

$l^{(5)}$

Mischen von Gesichtern

