Artistic Stylization of Images and Video

Part III – Anisotropy and Filtering

Eurographics 2011

Jan Eric Kyprianidis

Hasso-Plattner-Institut, University of Potsdam, Germany

Image/Video Abstraction

- **Stylized Augmented Reality for Improved Immersion**
  Fischer et al., 2005
- **Real-time Video Abstraction**
  Winnemöller et al., SIGGRAPH 2006
- **Coherent Line Drawings**
  Kang et al., NPAR 2007
- **Structure Adaptive Image Abstraction**
  Kyprianidis & Döllner, EG Theory and Practice of Computer Graphics 2008
- **Flow-based Image Abstraction**
  Kang et al., Transactions on Visualization and Computer Graphics 2009
- **Artistic Edge and Corner Preserving Smoothing**
  Papari et al., IEEE Transactions on Image Processing 2007
- **Image and Video Abstraction by Anisotropic Kuwahara Filtering**
  Kyprianidis et al., Pacific Graphics 2009
- **Shape-simplifying Image Abstraction**
  Kang & Lee, Pacific Graphics 2008
- **Image and Video Abstraction by Coherence-Enhancing Filtering**
  Kyprianidis & Kang, Eurographics 2011

Stylized Augmented Reality for Improved Immersion

Fischer et al. (2005)

Non-photorealistic display of both the camera image and virtual objects:

- **Abstraction:** Bilateral filter applied to Gaussian pyramid and then upsampled
- **Edges:** Canny edge detector + morphological dilation

Conventional augmented reality

Stylized augmented reality

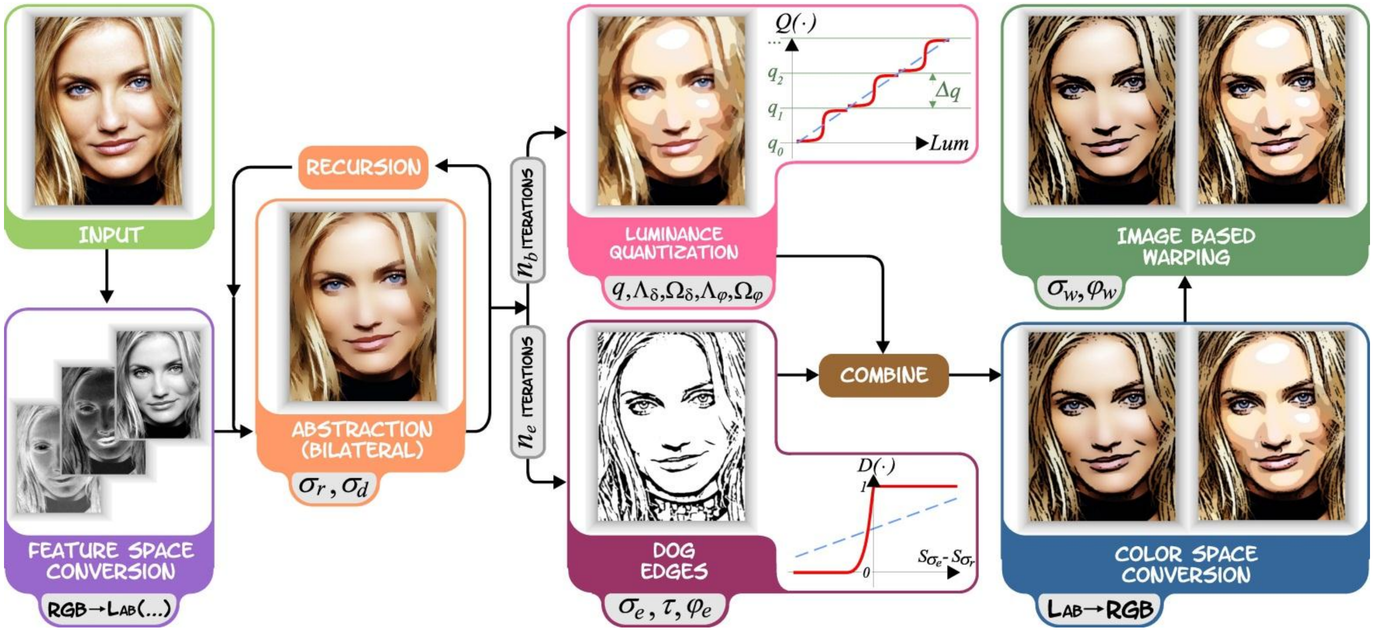
Image credit: Fischer et. al. (2005)

Real-time Video Abstraction

Winnemöller et al. (2006)

- **Abstraction:** Multiple iterations of xy-separable bilateral filter + color quantization
- **Edges:** Difference of Gaussians + thresholding

Image credit: Winnemöller et. al. (2006)



Coherent Line Drawings

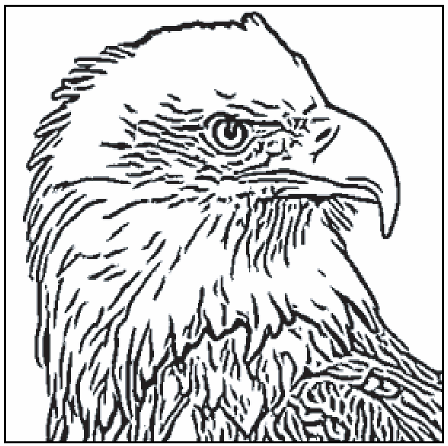
Kang et al. (2007)

- **Edges:** 1D difference of Gaussians directed by flow field + flow-guided smoothing and thresholding.

Image credit: Kang et. al. (2007)

Input image

Edge Tangent Flow



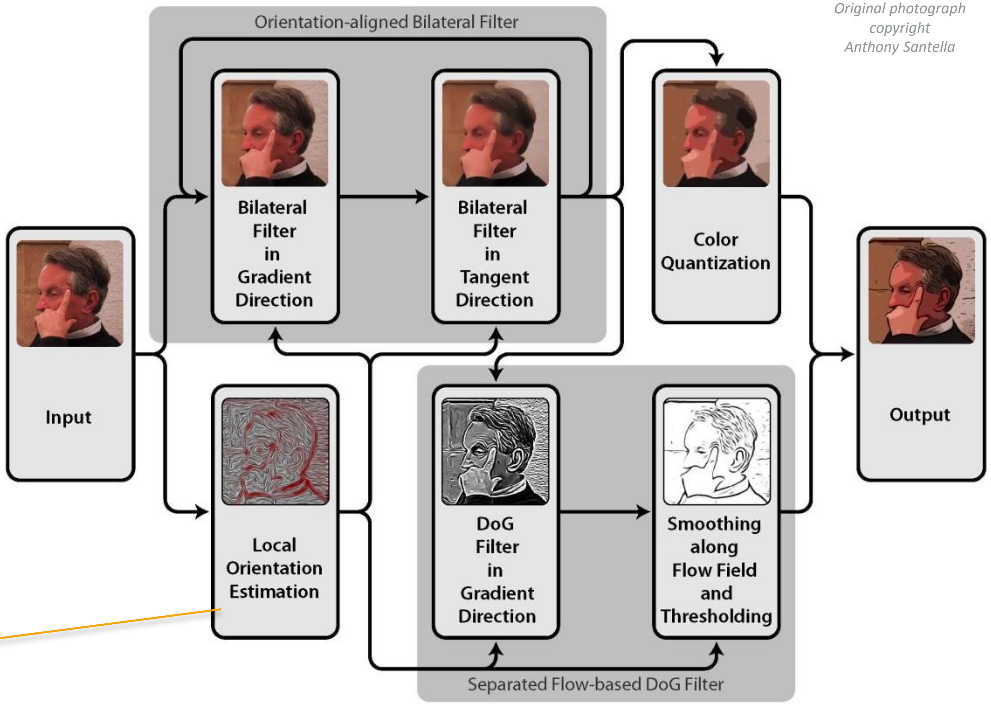
Line drawing

Flow-based filtering

Structure Adaptive Image Abstraction

Kyprianidis & Döllner (2008)

- **Abstraction:** Multiple iterations of orientation-aligned bilateral filter
- **Edges:** separable flow-based difference of Gaussians



Original photograph copyright Anthony Santella

Local orientation and an anisotropy measure derived from the smoothed structure tensor are used to guide the bilateral and difference of Gaussians filters

Flow-based Image Abstraction

Kang et al. (2009)

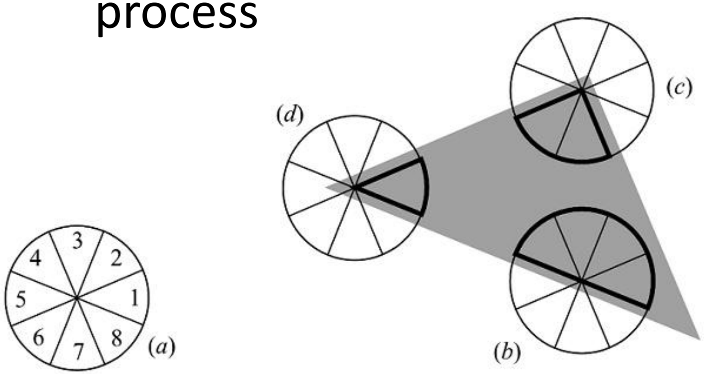
- **Abstraction:** Multiple iterations of flow-based bilateral filter
- **Edges:** (separable) flow-based difference of Gaussians
- Local orientation estimation of both techniques is based on the edge tangent flow (ETF)

Image credit: Kang et. al. (2009)

Artistic Edge and Corner Preserving Smoothing

Papari et al. (2009)

- Generalization of the Kuwahara filter. Creates output with a painterly look.
- Addresses two key issues of the original Kuwahara filter:
  - Rectangular subregions
  - Unstable subregion selection process

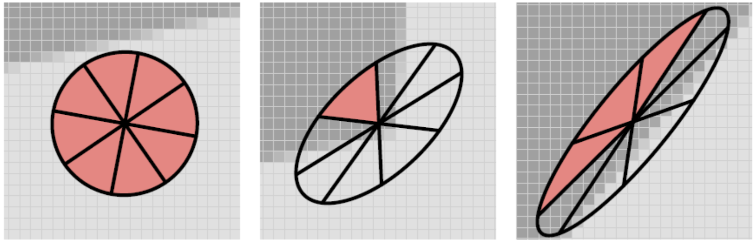


Credit for images: Papari et. al. (2009)

Anisotropic Kuwahara Filtering

Kyprianidis et al. (2009)

- Further generalization of the Kuwahara filter.
- Adaptation of the shape, scale and orientation of the filter to the local image structure.



Original image by Paulo Brandão@flickr.com

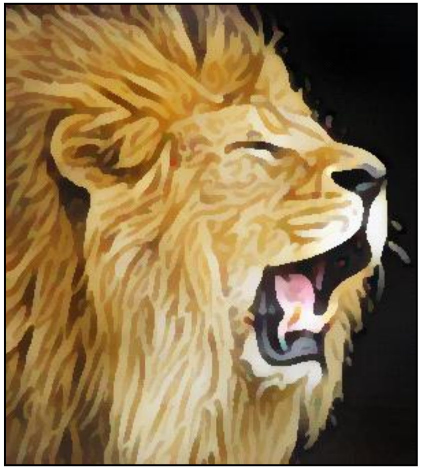
Shape-simplifying Image Abstraction

Kang & Lee (2008)

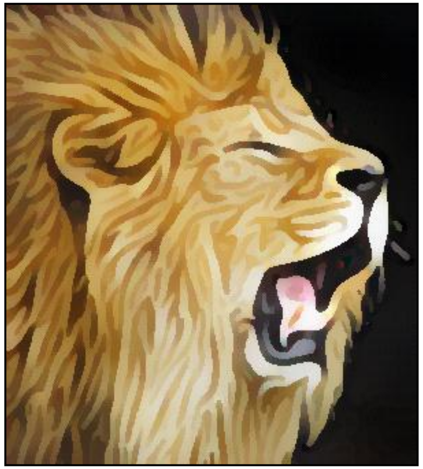
- PDE-based technique that simultaneous simplifies colors and shape:
  - Constrained mean curvature flow
  - Shock filter

Image credit: Kang & Lee (2008) / original image by Tambako the Jaguar@flickr.com

Input



20 iterations



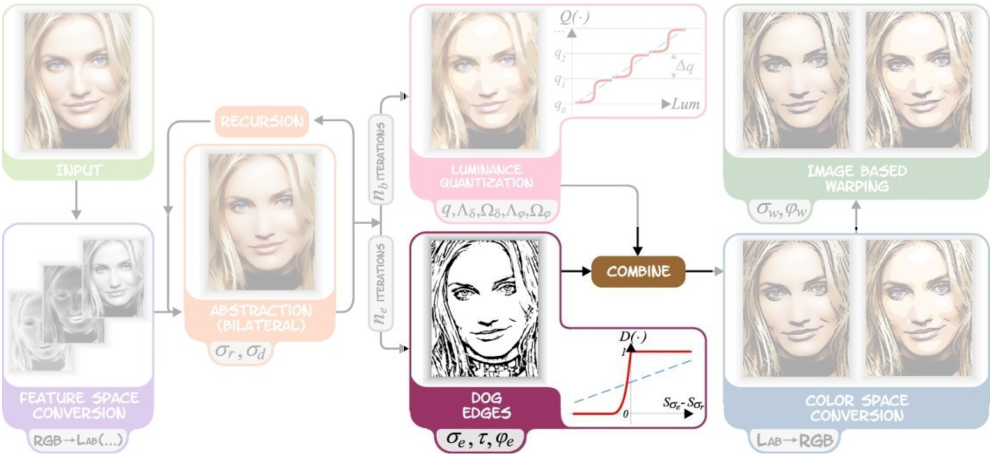
40 iterations

60 iterations

Difference of Gaussians

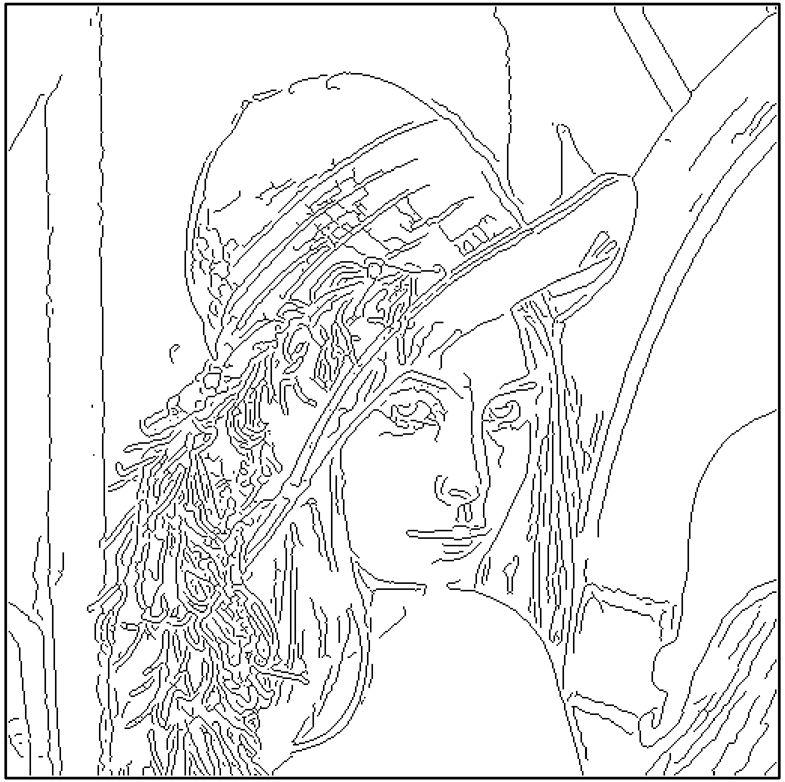
Difference of Gaussians:

- Laplacian of Gaussian (LoG)
- Isotropic Difference of Gaussians (DoG)
- Flow-based Difference of Gaussians
- Separable Flow-based Difference of Gaussians



DoG Edges vs Canny Edges

Original image from USC-SIPI Image Database



Canny Edges

Flow-based difference of Gaussians

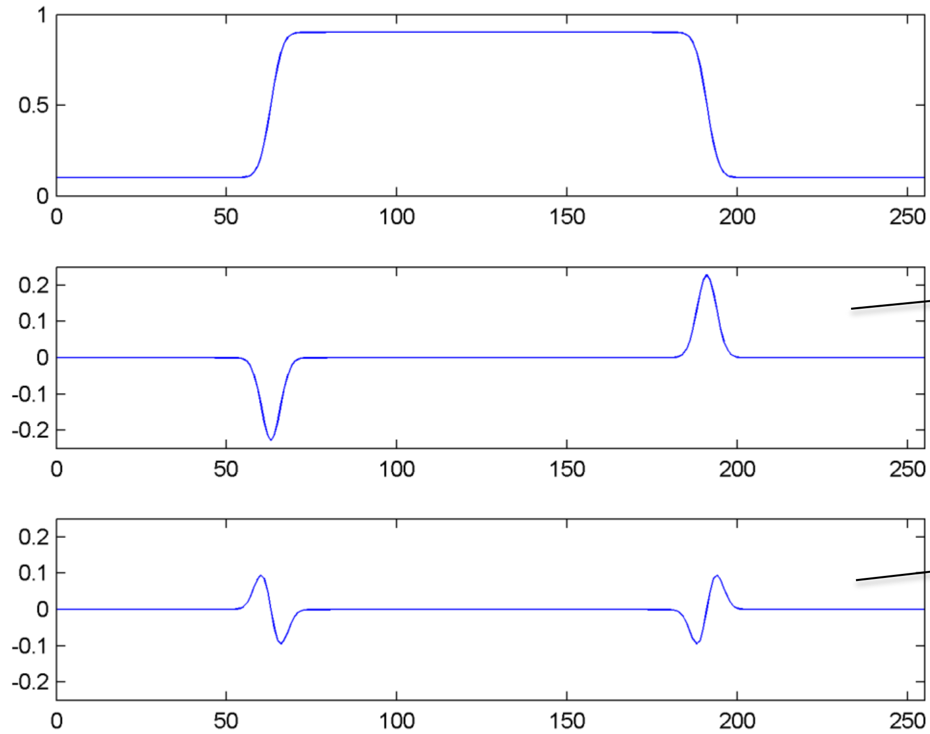
Edge Detection

Edge profile without noise:

Scanline of an image

First derivative of scanline

Second derivative of scanline



Edges are located at minima or maxima of the first derivative

Edges are located at zero crossings of the second derivative

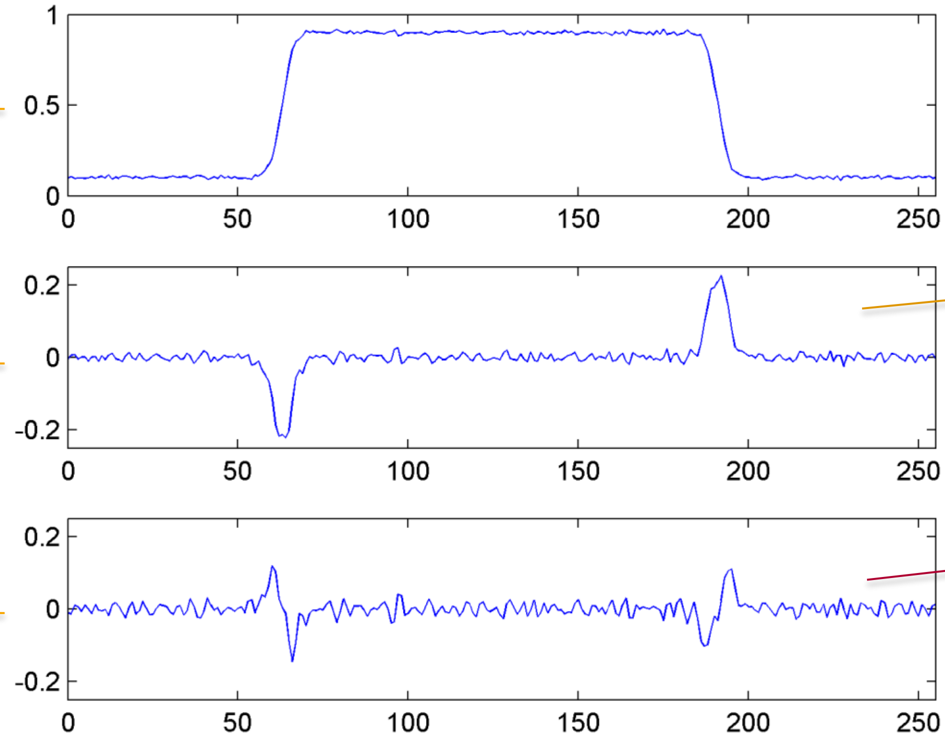
Edge Detection

Edge profile with noise:

Scanline of an image **with noise**

First derivative of scanline

Second derivative of scanline



First derivative sensitive to noise

Second derivative even more sensitive to noise

Laplacian of Gaussian (LoG)

Marr & Hildreth (1980)

In 2D the second derivative corresponds to the Laplacian:

$$L = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

Similar to the second derivative the Laplacian is sensitive to noise. To make the Laplacian less sensitive to noise, apply a Gaussian to the image first:

$$\mathrm{LoG} = L \star G_\sigma \ ,$$

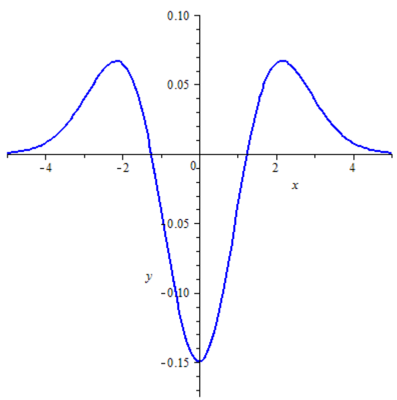
where G_σ is a 2D Gaussian with standard deviation σ:

$$G_\sigma(x, y) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{x^2 + y^2}{2\sigma^2} \right)$$

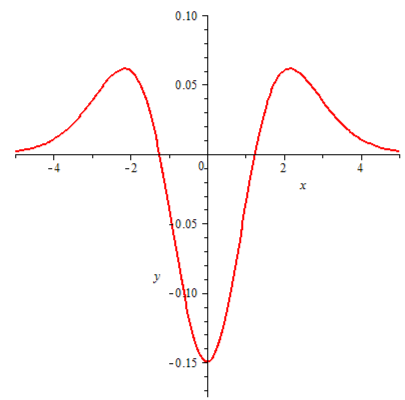
Difference of Gaussians (DoG)

Marr & Hildreth (1980)

A Laplacian of Gaussian can be approximated by a difference of Gaussians:



$$\mathrm{LoG}(x,y) = L \star G_\sigma$$



$$\mathrm{DoG}(x,y) = G_{\sigma_i}(x,y) - G_{\sigma_e}(x,y)$$

Good engineering solution:
$\sigma_i = 1.6 \cdot \sigma_e$

Fast to implement since Gaussian is separable

Difference of Gaussians (DoG)

Marr & Hildreth (1980)

Zero-crossing are found by thresholding:

Smooth DoG Edges

Winnemöller et al. (2006)

An approach to created smooth edges was proposed by Winnemöller et al.:

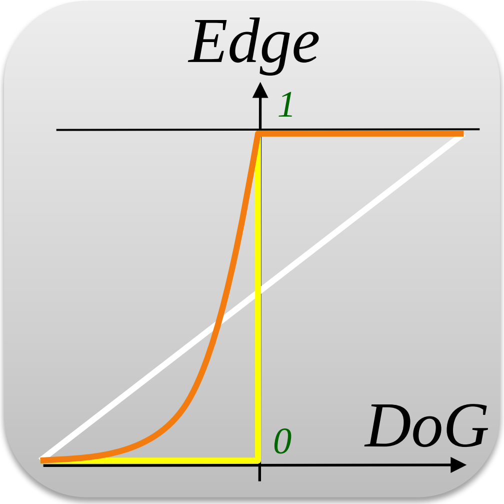
$$D(\sigma_e, \tau, \varphi_e) = \begin{cases} 1 & \text{if } \left(G_{\sigma_e} - \tau\, G_{1.6\cdot\sigma_e}\right) > 0 \\ 1 + \tanh(\varphi_e \cdot G_{\sigma_e} - \tau G_{1.6\cdot\sigma_e}) & \text{otherwise} \end{cases}$$

- The parameter τ controls the sensitivity to noise. A typical values are $\tau = 0.98$ or $\tau = 0.99$.
- The falloff parameter φ_e determines the sharpness of edge representations, typical values are $\varphi_e \in [0.75, 5.0]$.

Smooth DoG Edges

Winnemöller et al. (2006)

tanh
(half-truncated)



Credit for slide: H. Winnemöller

Difference of Gaussians

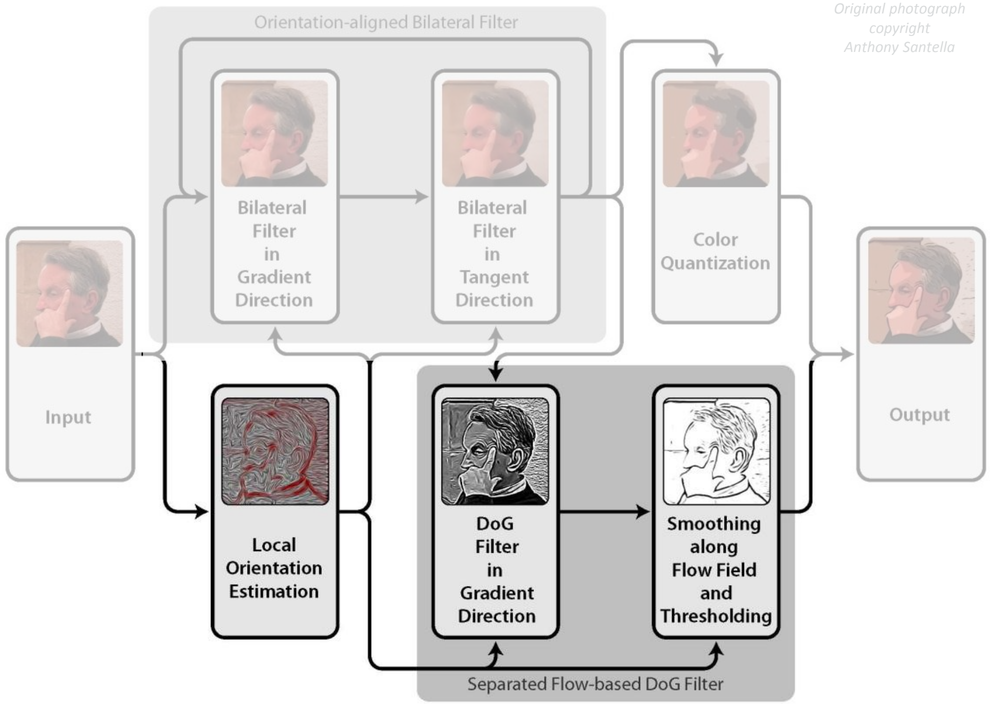
Guided by Local Image Structure

Local Structure Estimation:

- Edge Tangent Flow
- Structure Tensor

DoG Guided by Local Image Structure:

- Flow-based DoG
- Separable flow-based DoG

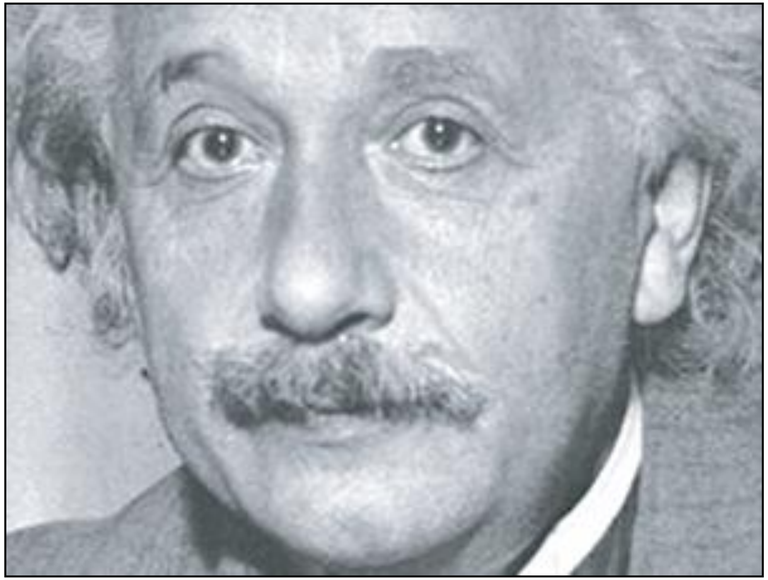


Edge Tangent Flow
Kang et al. (2007)

Edge Tangent Flow (ETF):

- Smoothly varying vector field
- Feature-preserving flow

Image credit: Kang et al. (2007)



Input image

Edge Tangent Flow

Edge Tangent Flow

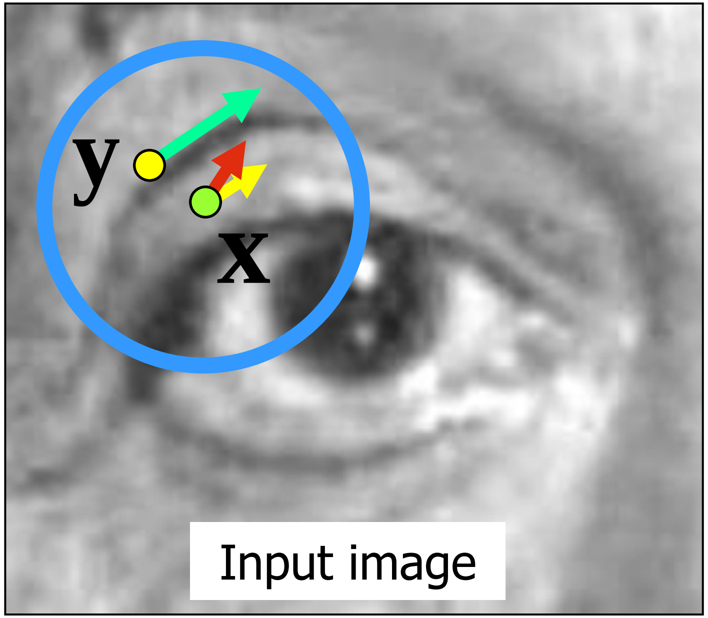
Kang et al. (2007)

Weighted vector smoothing similar to bilateral filter:

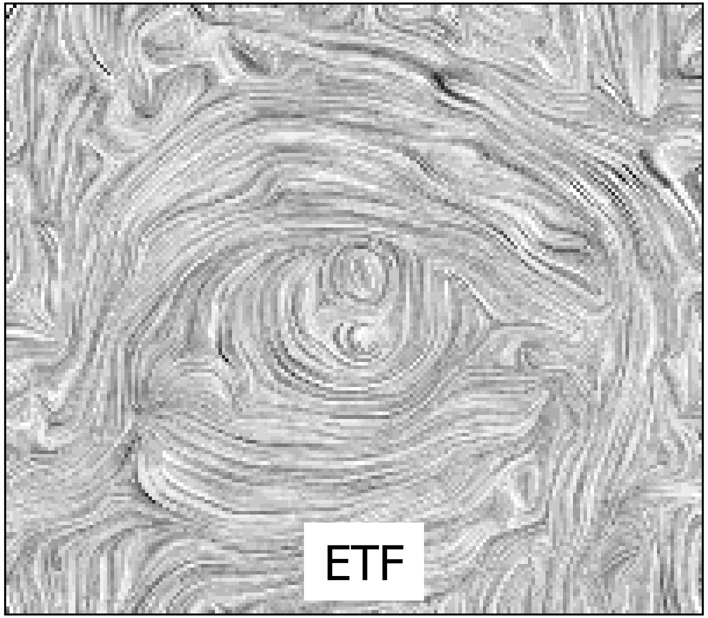
Multiple iterations (≈ 3)

$$t^{n+1}(x) = \frac{1}{k} \sum_{y \in \Omega(x)} \phi(x,y) \cdot t^n(y) \cdot w_s(x,y) \cdot w_m(x,y) \cdot w_d(x,y)$$

t^0 is calculated using the Sobel filter.



Input image



ETF

Image credit: Kang et al. (2007)

Edge Tangent Flow

Kang et al. (2007)

$$t^{n+1}(x) = \frac{1}{k} \sum_{y \in \Omega(x)} \phi(x,y) \cdot t^n(y) \cdot w_s(x,y) \cdot w_m(x,y) \cdot w_d(x,y)$$

Assure different vectors point in the same direction

$$\phi(x,y) = \operatorname{sign}(t^n(x) \cdot t^n(y))$$

$$w_s(x,y) = \begin{cases} 1 & |x-y| < r \\ 0 & \text{else} \end{cases}$$

Restrict filtering to a predefined radius

More weight to vectors with higher gradient magnitude

$$w_m(x,y) = \frac{1}{2}[1 + \tanh(|g(x)| - |g(y)|)]$$

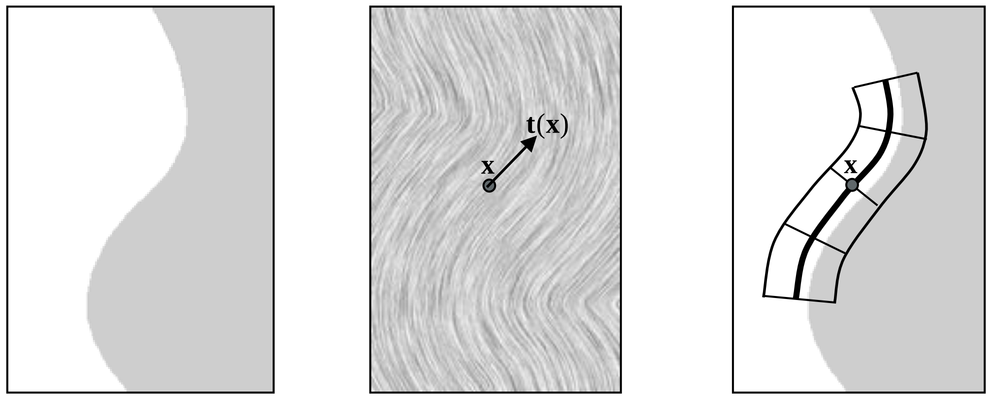
$$w_d(x,y) = |t^n(x) \cdot t^n(y)|$$

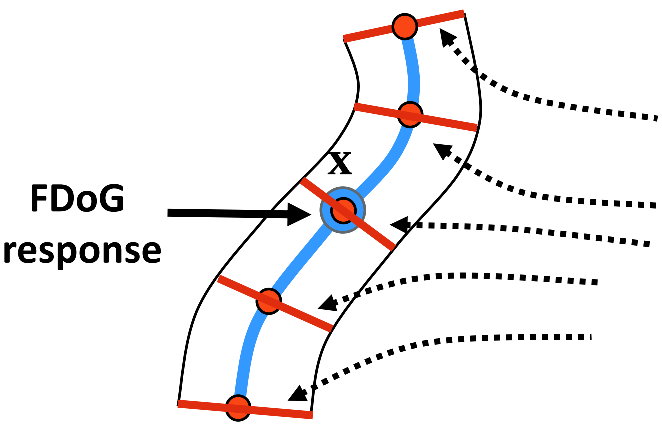
More weight for vectors with direction similar to current filter origin

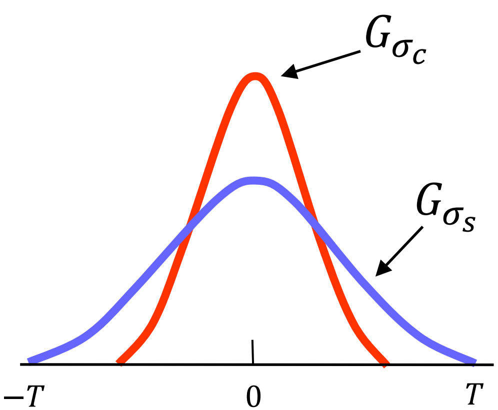
Flow-based Difference of Gaussians

Kang et al. (2007)

Image credit: Kang et al. (2007)







DoG filter

Structure Tensor

Let $f: \mathbb{R}^2 \to \mathbb{R}^3$ denote the input image and let

$$\frac{\partial f}{\partial x} = \begin{pmatrix} \frac{\partial R}{\partial x} & \frac{\partial G}{\partial x} & \frac{\partial B}{\partial x} \end{pmatrix}^t \qquad \frac{\partial f}{\partial y} = \begin{pmatrix} \frac{\partial R}{\partial y} & \frac{\partial G}{\partial y} & \frac{\partial B}{\partial y} \end{pmatrix}^t$$

These can be implemented for example using Gaussian derivatives or the Sobel filter.

be the partial derivatives of f.

The structure tensor is then defined by:

$$(g_{ij}) = J^t J = \begin{pmatrix} \left\langle \frac{\partial f}{\partial x}, \frac{\partial f}{\partial x} \right\rangle & \left\langle \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right\rangle \\ \left\langle \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right\rangle & \left\langle \frac{\partial f}{\partial y}, \frac{\partial f}{\partial y} \right\rangle \end{pmatrix} =: \begin{pmatrix} E & F \\ F & G \end{pmatrix}$$

The structure tensor is a 2×2 symmetric positive semidefinte matrix

In differential geometry the structure tensor is also known as first fundamental form

Structure Tensor

The induced quadratic form of the structure tensor measures the squared rate of change of f in direction of a vector $n = (n_x, n_y)$:

$$S(n) = En_x^2 + 2Fn_xn_y + Gn_y^2$$

The extremal values of $S(n)$ on the unit circle correspond to the eigenvalues of (g_{ij}):

$$\lambda_{1,2} = \frac{E + G \pm \sqrt{(E-G)^2 + 4F^2}}{2}$$

The corresponding eigenvectors are:

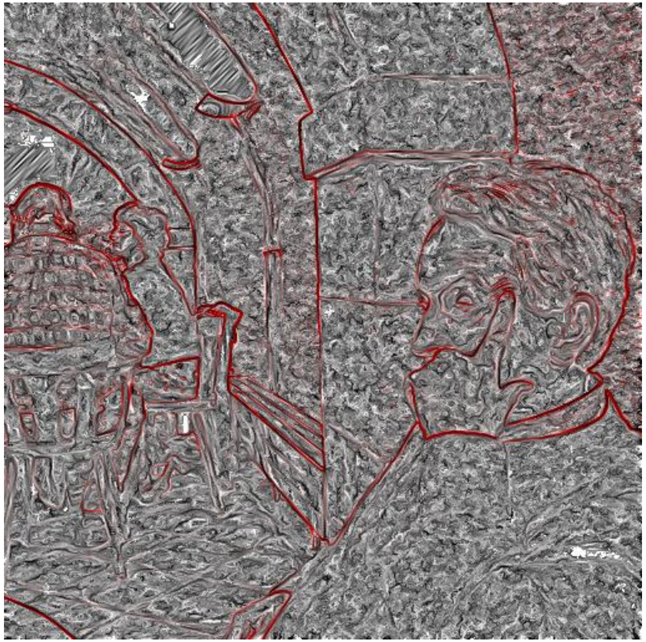
Eigenvector of major eigenvalue. Direction of maximum change: **gradient** direction.

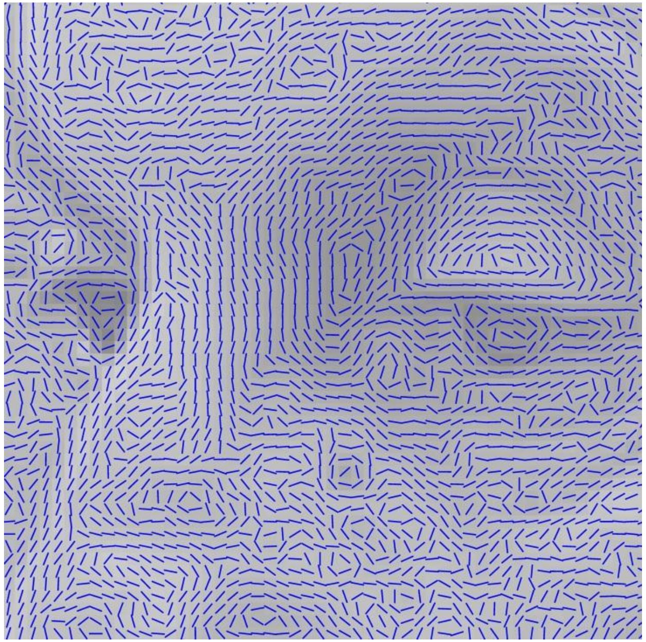
$$v_1 = \begin{pmatrix} F \\ \lambda_1 - E \end{pmatrix} \qquad v_2 = \begin{pmatrix} \lambda_1 - E \\ -F \end{pmatrix}$$

Eigenvector of minor eigenvalue. Direction of minimum change: **tangent** direction.

Structure Tensor

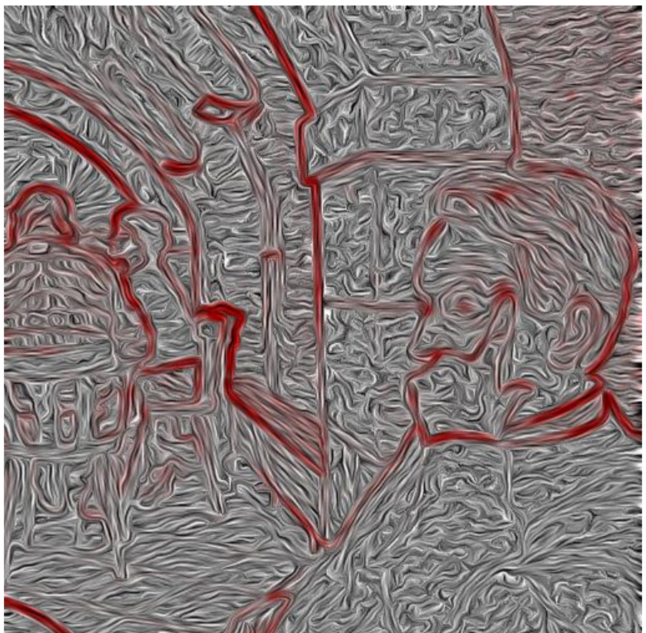
The eigenvectors corresponding to the minor eigenvalues of the structure define a vector field. Typically this field is not smooth:

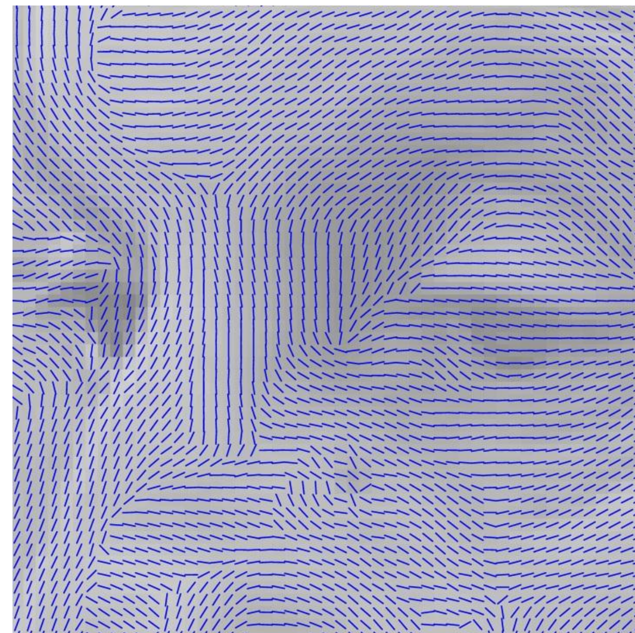




Smoothed Structure Tensor

Smoothing the structure tensor prior to eigenanalysis with a Gaussian filter removes discontinuities in the vector field:





Smoothed Structure Tensor

Eigenvector field of the smoothed structure tensor is similar to the edge tangent flow, but allows a more efficient implementation:

3 iterations of edge tangent flow filter

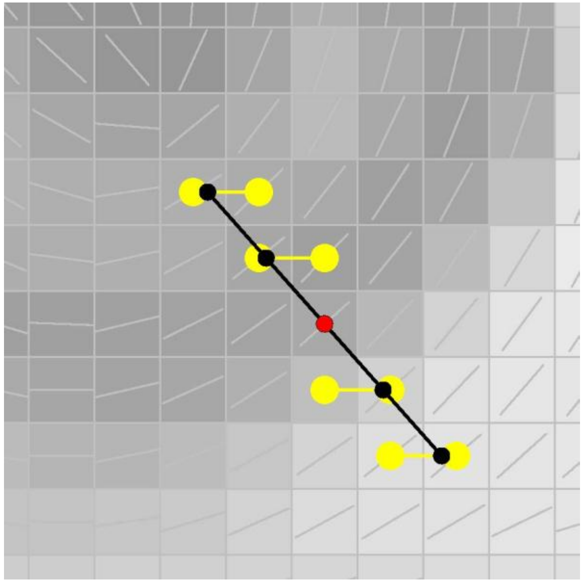
Eigenvector field of the smoothed structure tensor

Separable Flow-based Difference of Gaussians

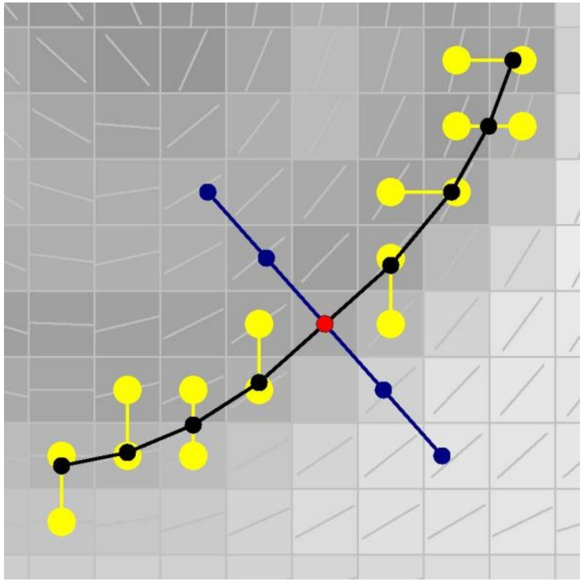
Kyprianidis & Döllner (2008)

Split flow-based difference of Gaussians into two passes:

- 1st Pass: one-dimensional DoG in direction of the major eigenvector
- 2nd Pass: smoothing along stream lines defined by minor eigenvector



1st Pass

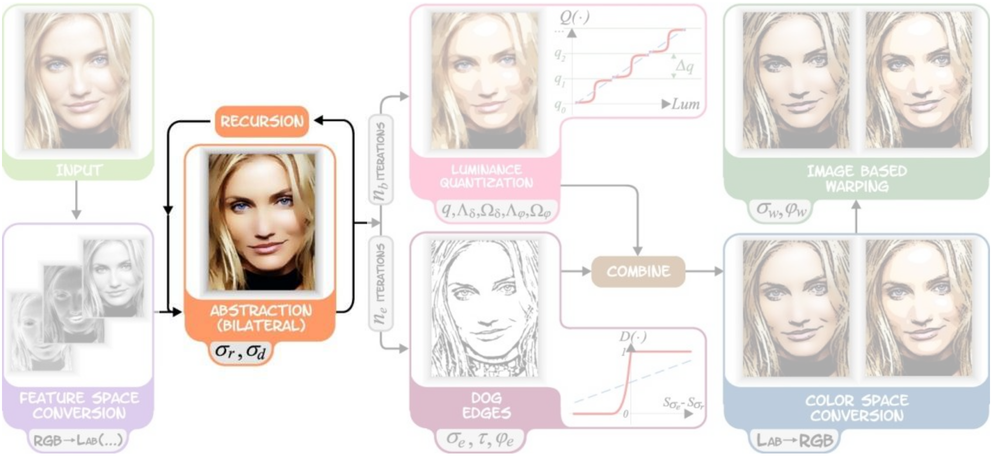


2nd Pass

Bilateral Filter

Bilateral Filter:

- Classical Bilateral Filter
- xy-Separable Bilateral Filter
- Orientation-aligned Bilateral Filter
- Flow-based Bilateral Filter



Bilateral Filter

Tomasi & Manduchi (1998)

The bilateral filter is a nonlinear operation that smoothes images while preserving edges:

More weight to closer pixel

More weight to pixel with similar color

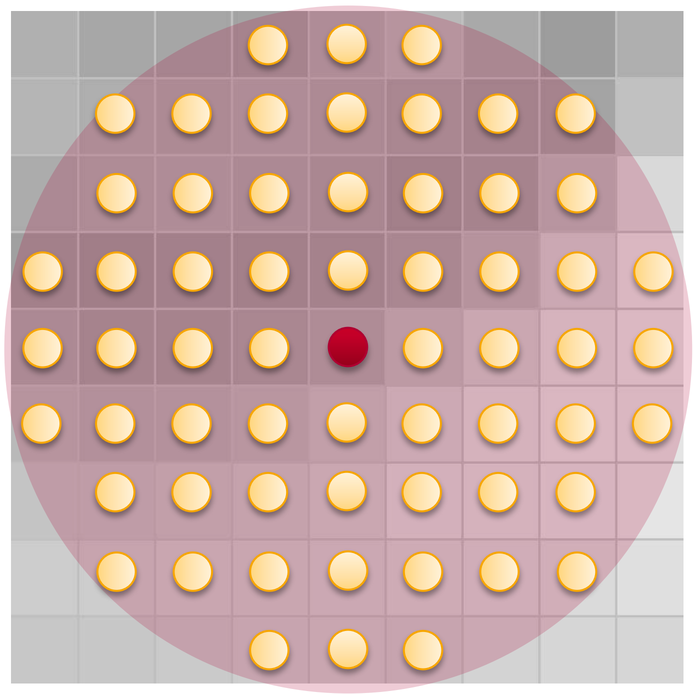
$$\frac{\sum_{x \in B_r(x_0)} f(x)\, \boxed{G_{\sigma_d}(|x - x_0|)}\, \boxed{G_{\sigma_r}(|f(x) - f(x_0)|)}}{\sum_{x \in B_r(x_0)} G_{\sigma_d}(|x - x_0|)\, G_{\sigma_r}(|f(x) - f(x_0)|)}$$

$$G_\sigma(t) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{t^2}{2\sigma^2}\right)$$

Bilateral Filter

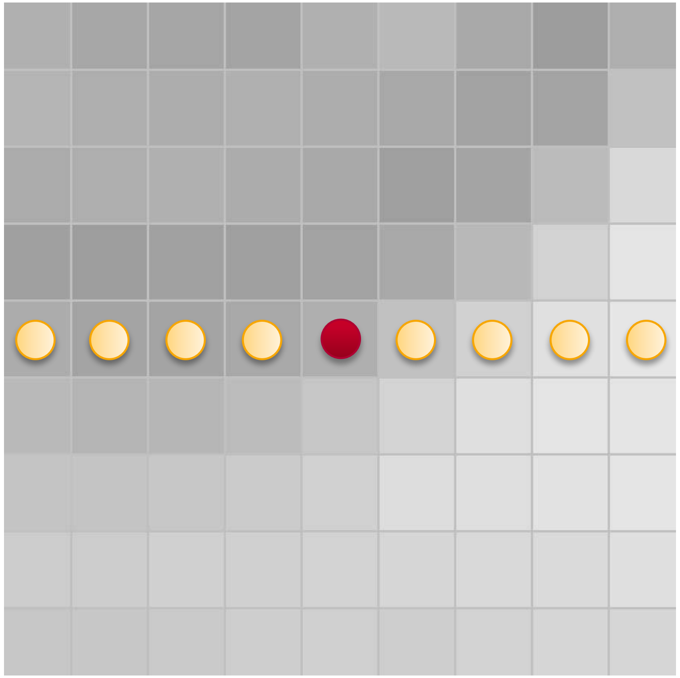
Tomasi & Manduchi (1998)

The bilateral filter is a powerful tool, but computationally very expensive ($O(r^2)$ per pixel).

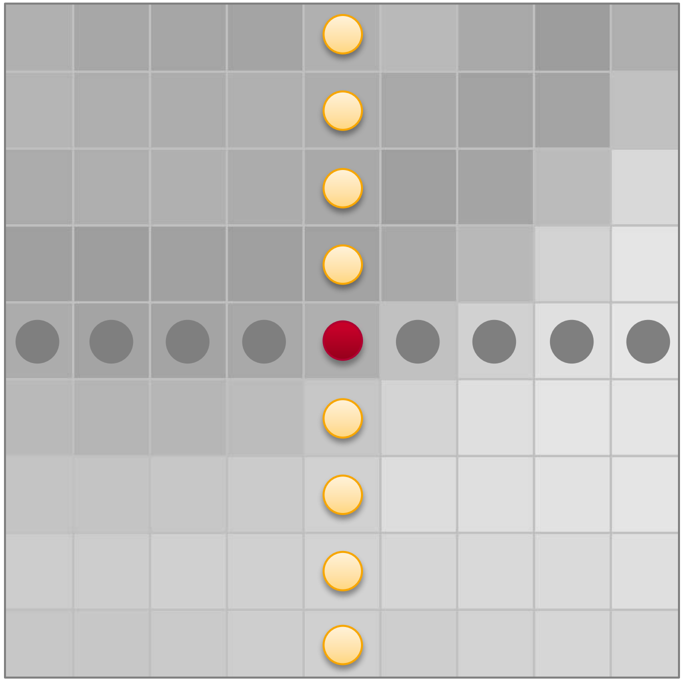


xy-Separable Bilateral Filter

Pham & van Vliet (2005)



1st Pass

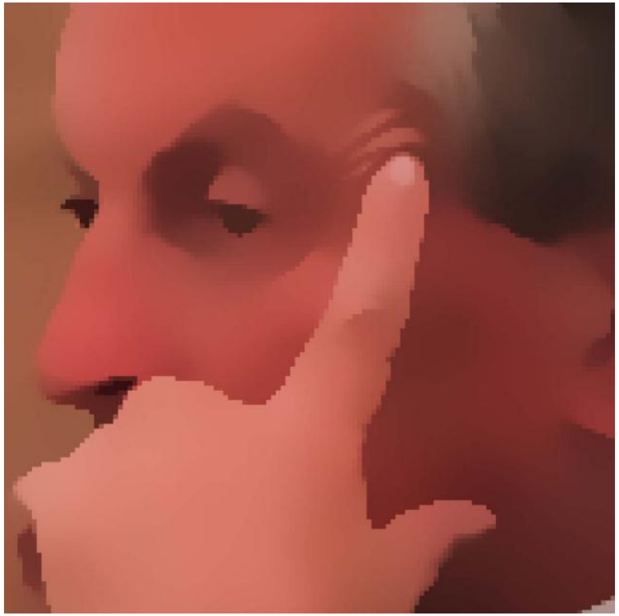


2nd Pass

xy-Separable Bilateral Filter

Pham (2006)

- Much faster than classical bilateral filter
- But creates noticeable artifacts!



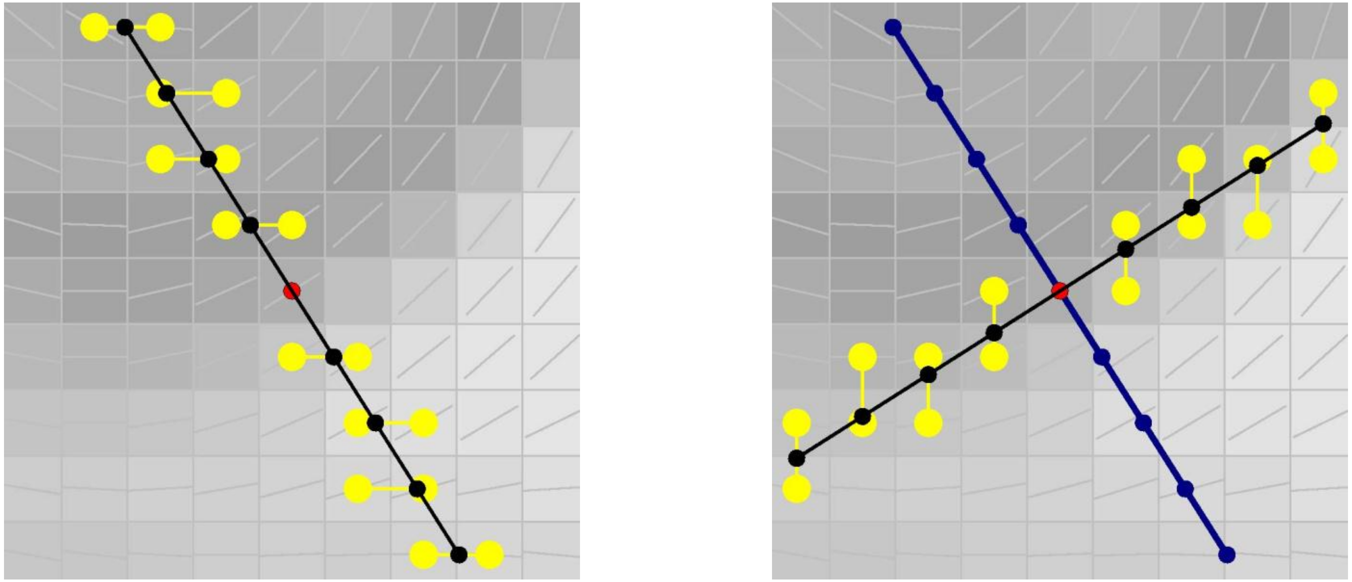
Full kernel bilateral filter

xy-separable bilateral filter

Orientation-aligned Bilateral Filter

Kyprianidis & Döllner (2008)

- Align separable bilateral filter to local orientation derived from the smoothed structure tensor

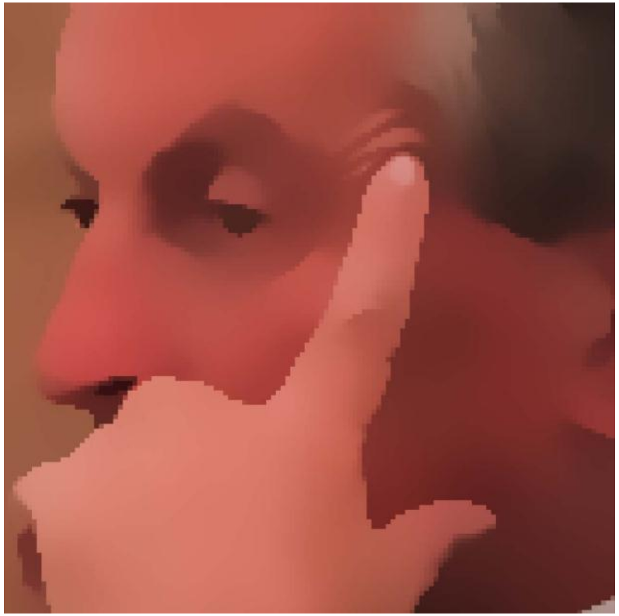


1st Pass 2nd Pass

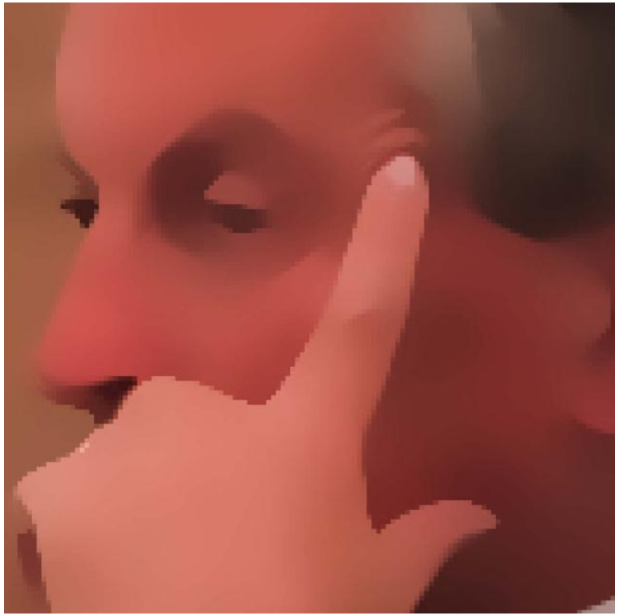
Orientation-aligned Bilateral Filter

Kyprianidis & Döllner (2008)

- Less artifacts. Very well suited for abstraction.



Full kernel bilateral filter



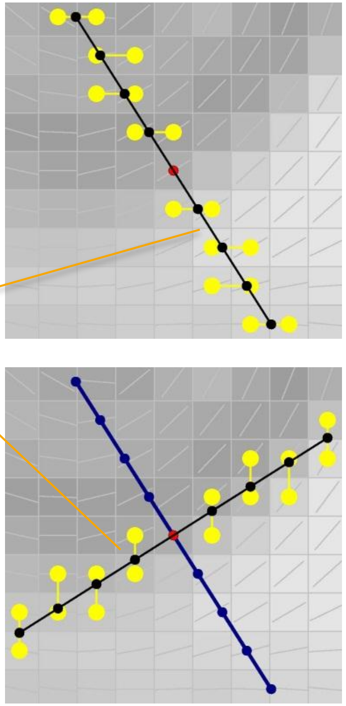
Orientation-aligned bilateral filter

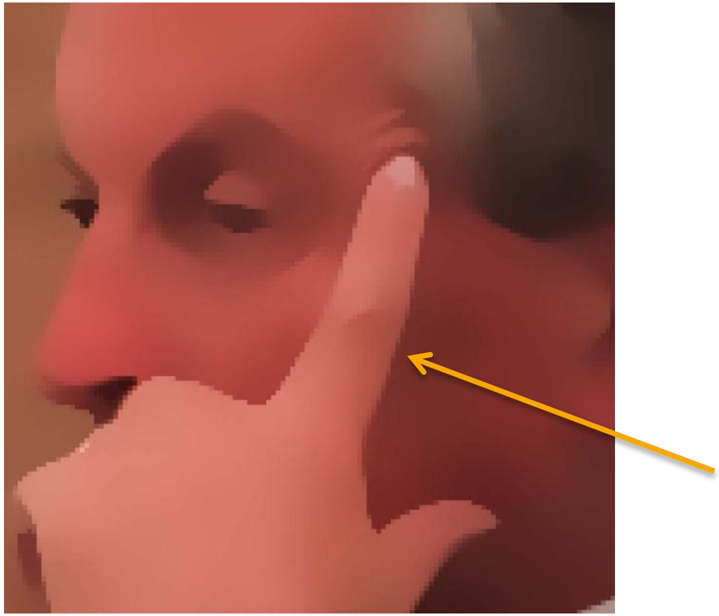
Orientation-aligned Bilateral Filter

Kyprianidis & Döllner (2008)

- Linear smoothing of neighboring pixel values creates smooth color boundaries

Unit step size either along x- or y-axis



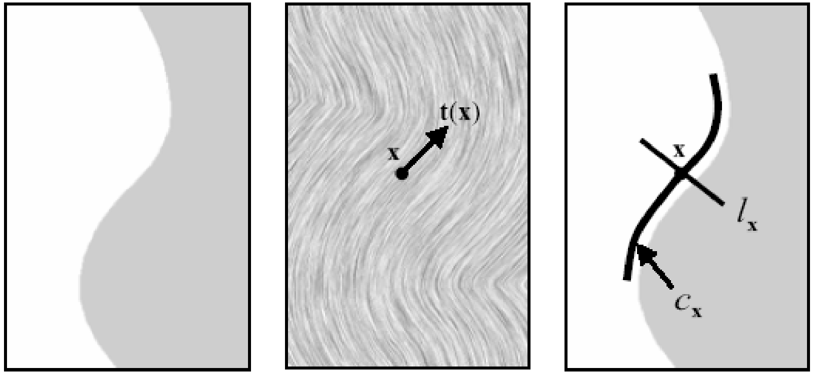


Flow-based Bilateral Filter

Kang et al. (2009)

Image credit: Kang et al. (2009)

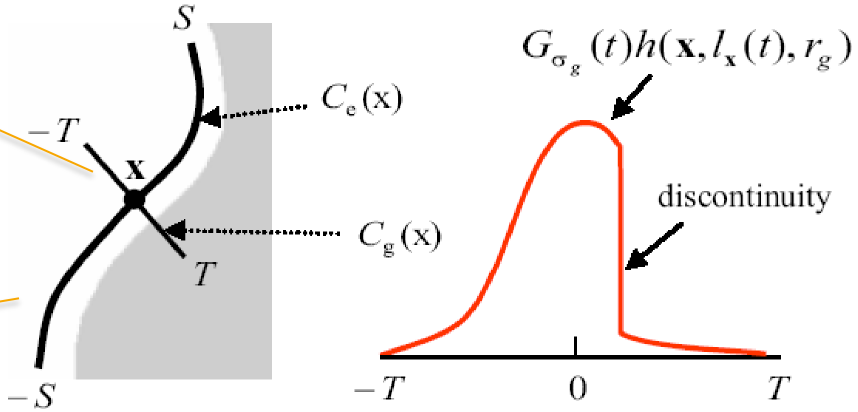
- ETF-guided decomposition of bilateral filter



(a) (b) (c)

1st pass in direction of the tangent

2nd pass along stream line defined by the ETF



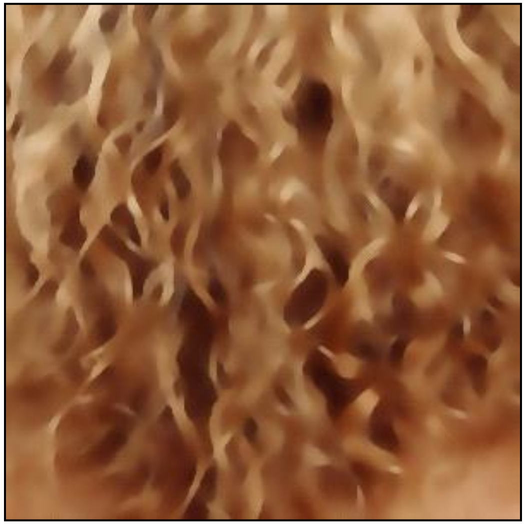
Flow-based Bilateral Filter

Kang et al. (2009)

- Excellent preservation of highly anisotropic image features

Image credit: Kang et al. (2009)

Input image

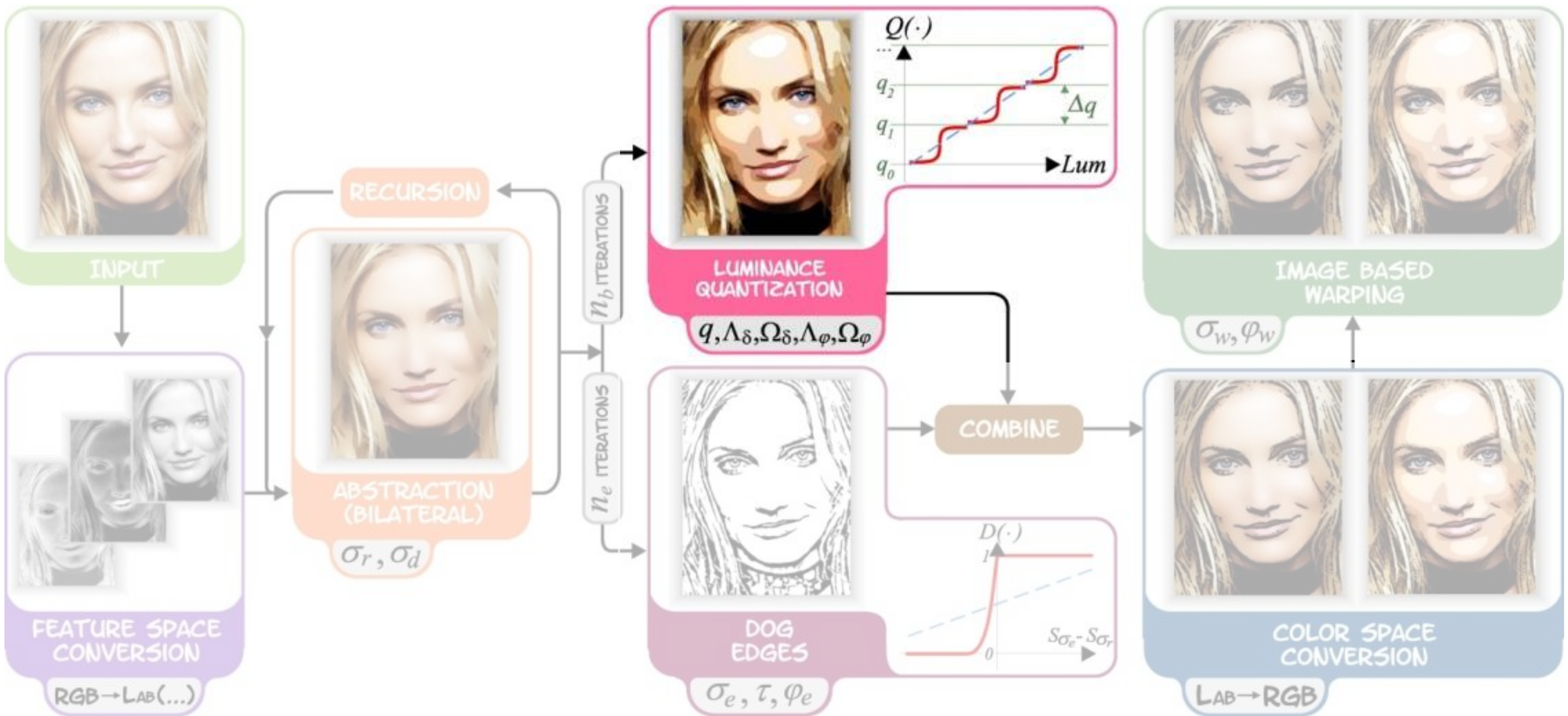


Bilateral filter

Flow-based bilateral filter

Color Quantization

Winnemöller et al. (2006)

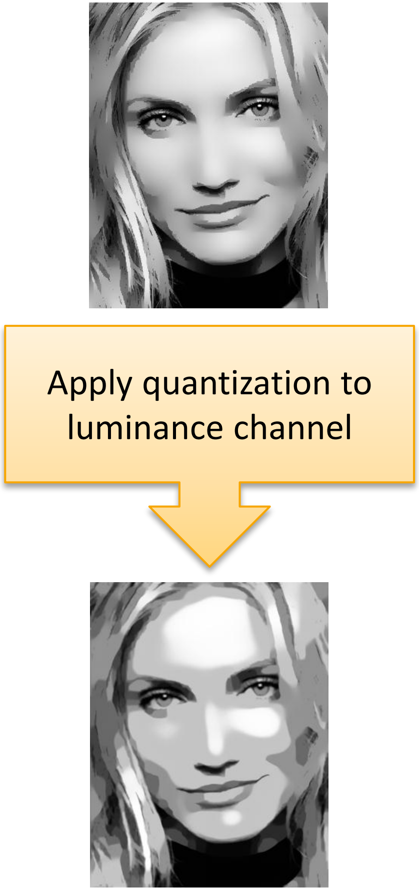


Color Quantization

Winnemöller et al. (2006)

Credit for slide: H. Winnemöller

Input



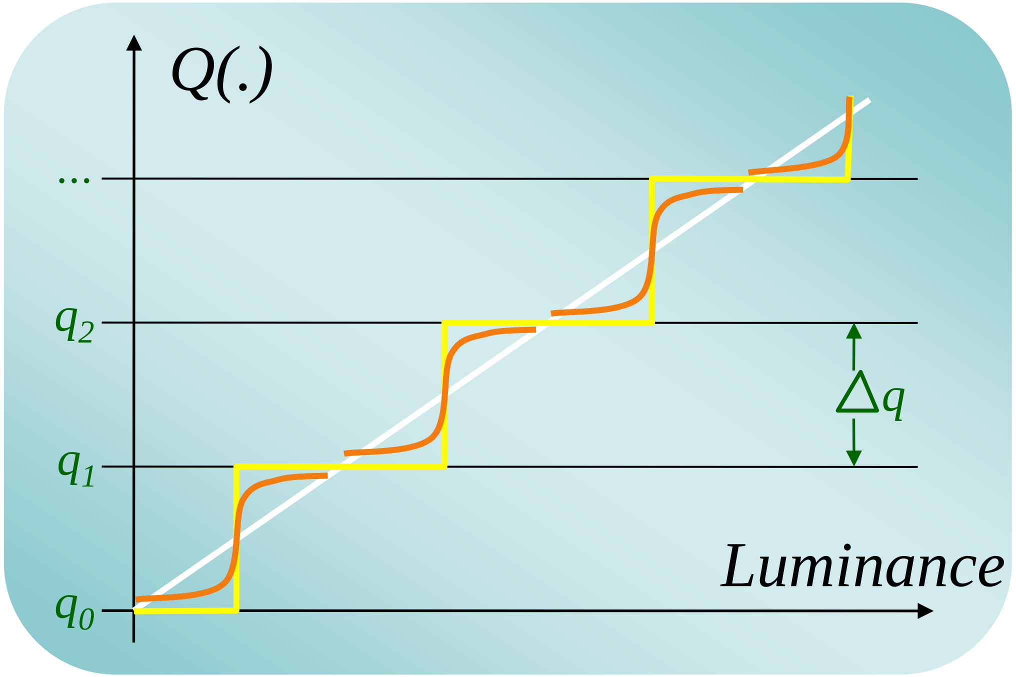
Result

Color Quantization

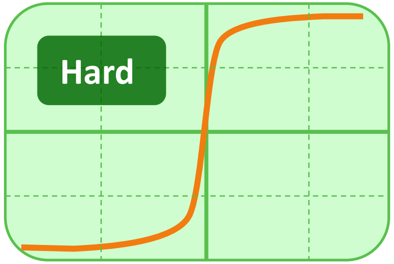
Winnemöller et al. (2006)

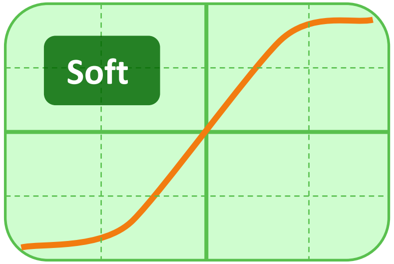
Luminance Mapping

q **bins**



tanh
per bin





Credit for slide: H. Winnemöller

Color Quantization

Winnemöller et al. (2006)

Image credit: Winnemöller et al. (2006)

Abstracted

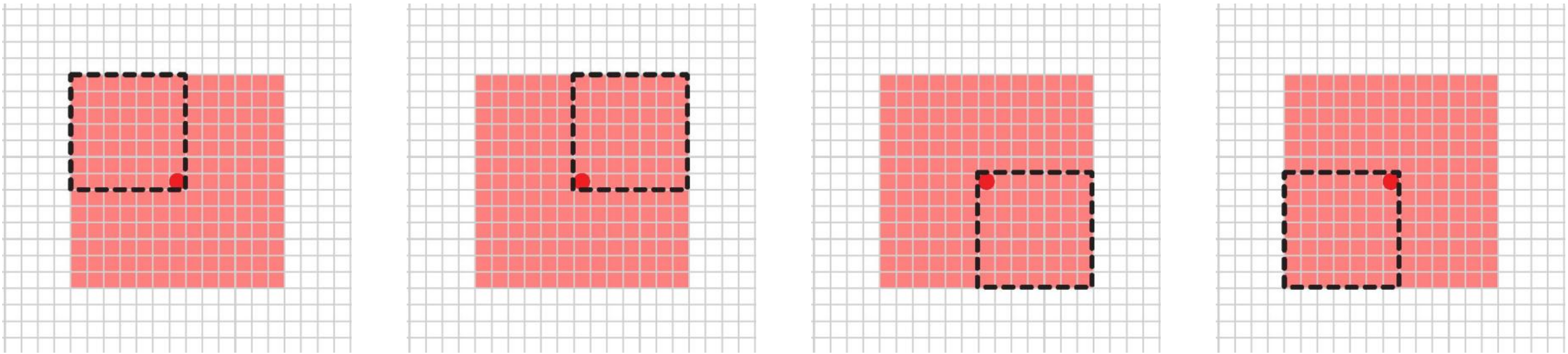
Sharp Quantization
(*Toon*-like)

Smooth Quantization
(*Paint*-like)

Kuwahara Filter:

- Classical Kuwahara Filter
- Kuwahara Filter with Weighting Functions
- Generalized Kuwahara Filter
- Anisotropic Kuwahara Filter
  - Convolution-based Weighting Functions
  - Polynomial Weighting Functions

Kuwahara Filter



$$
\begin{aligned}
W_0 &= [x_0 - r, x_0] \times [y_0, y_0 + r] \\
W_1 &= [x_0, x_0 + r] \times [y_0, y_0 + r] \\
W_2 &= [x_0, x_0 + r] \times [y_0 - r, y_0] \\
W_3 &= [x_0 - r, x_0] \times [y_0 - r, y_0]
\end{aligned}
$$

Kuwahara Filter

For every subregion W_i calculate the mean

$$m_i = \frac{1}{|W_i|} \sum_{(x,y) \in W_i} I(x,y)$$

and the variance:

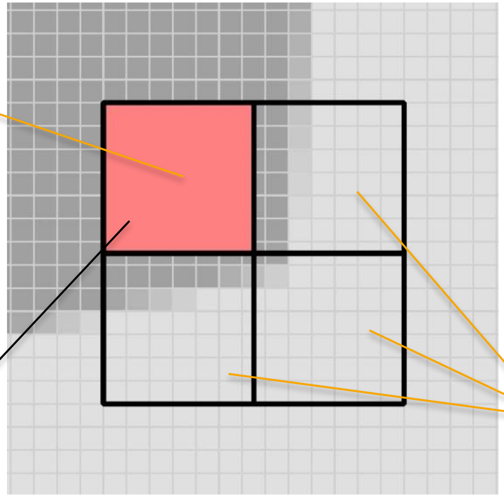
$$s_i^2 = \frac{1}{|W_i|} \sum_{(x,y) \in W_i} (I(x,y) - m_i)^2$$

The output of the Kuwahara filter is then defined as the mean of a subregion with minimum variance:

$$F(x_0, y_0) := m_k, \qquad k = \underset{i=0,\dots,3}{\operatorname{argmin}} s_i$$

Kuwahara Filter

Kuwahara filter for a corner



Region with **small variance**. All pixels of this region are similar.

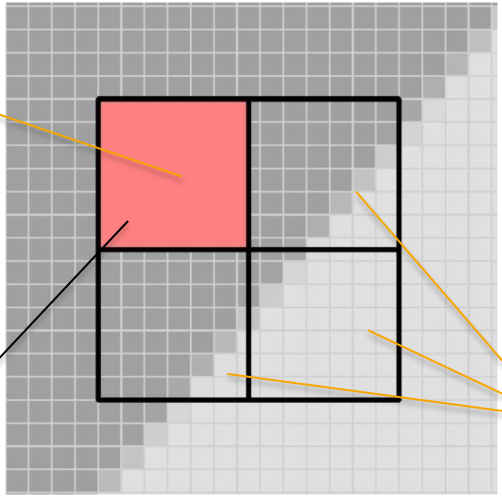
Result is the mean of the region with minimum variance

Regions with **high variance**. They all contain pixels from both color regions.

Kuwahara Filter

Kuwahara filter for an edge

Region with **small variance**. All pixels of the region lie on the same side of the edge.



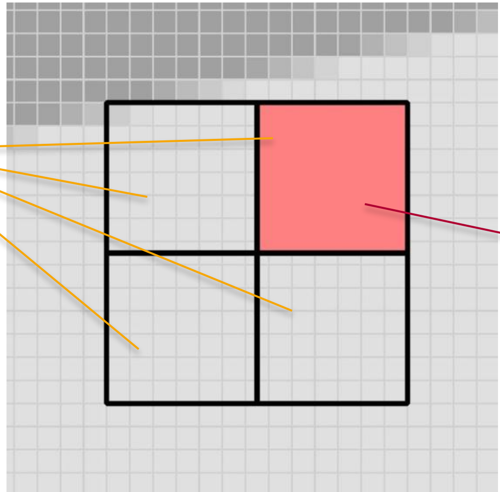
Result is the mean of the region with minimum variance

Regions with **high variance**. They all contain pixels from both sides of the edge.

Kuwahara Filter

Kuwahara filter for an homogenous neighborhood

All regions have **similar variance**!

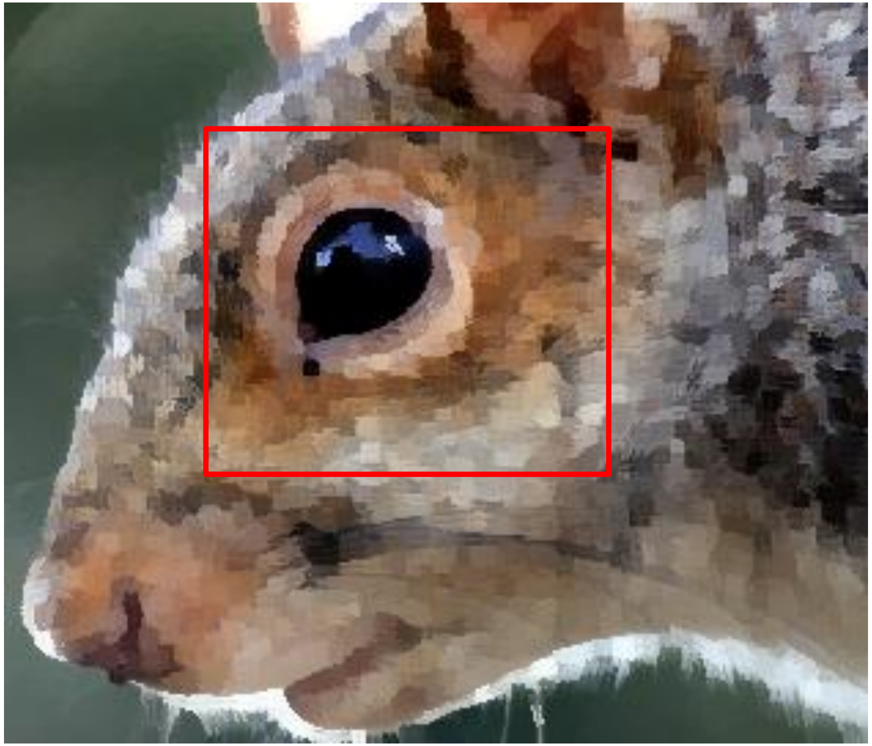


Selection of the region with minimum variance is highly unstable. For example if noise is present.

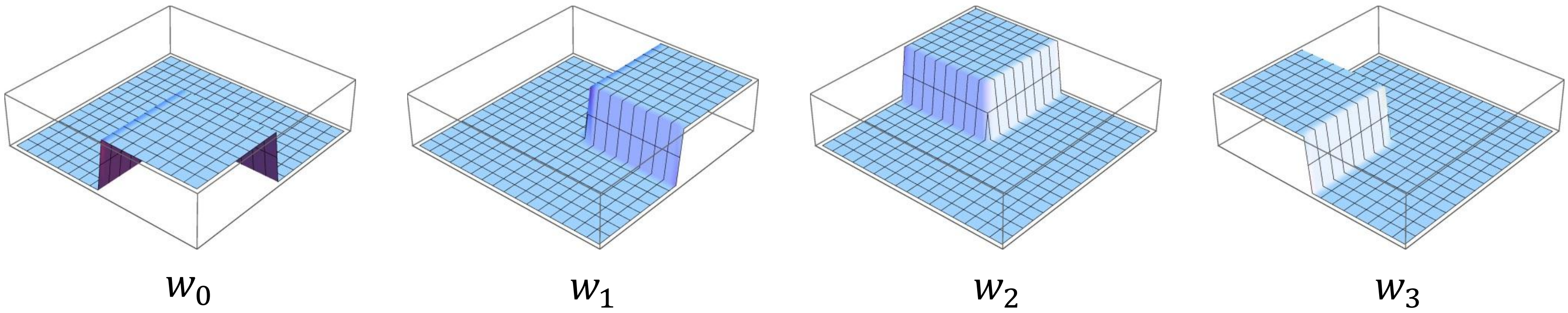
Kuwahara Filter

Original image by Keven Law@flickr.com

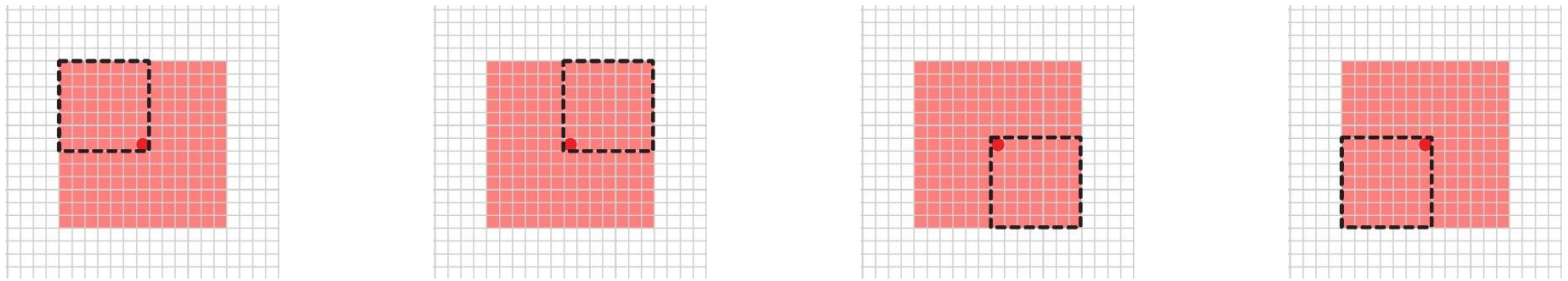
Kuwahara Filter



Kuwahara Filter with Weighting Functions



$$w_i(x,y) = \begin{cases} 1 & \text{if } (x,y) \in W_i \\ 0 & \text{otherwise} \end{cases}$$



Kuwahara Filter with Weighting Functions

Then the mean is given by:

$$m_i = \frac{1}{|W_i|} \sum_{(x,y)\in W_i} I(x,y)$$
$$= \frac{1}{|w_i|} \sum_{(x,y)\in \mathbb{Z}^2} I(x,y) \cdot w_i(x - x_0, y - y_0)$$

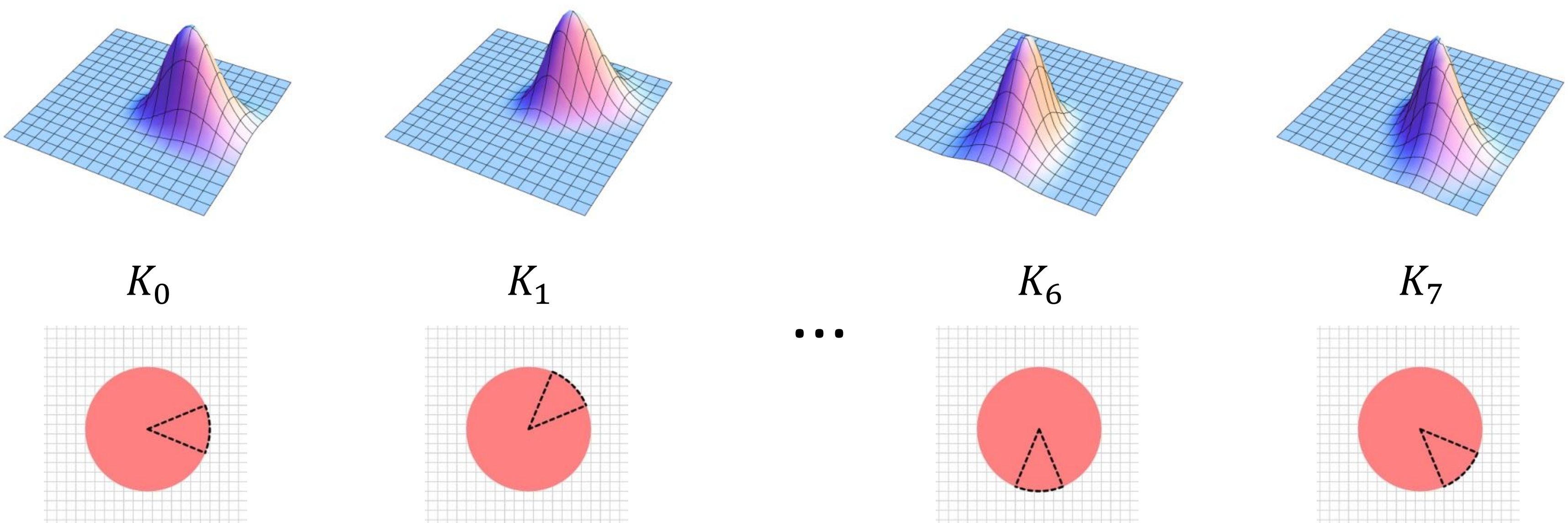
And the variance is given by:

$$s_i^2 = \frac{1}{|W_i|} \sum_{(x,y)\in W_i} (I(x,y) - m_i)^2$$
$$= \frac{1}{|w_i|} \sum_{(x,y)\in \mathbb{Z}^2} (I(x,y) - m_i)^2 \cdot w_i(x - x_0, y - y_0)$$

Generalized Kuwahara Filter

Papari et al. (2007)

Idea: Create smooth weighting functions over a disc those sum is a Gaussian

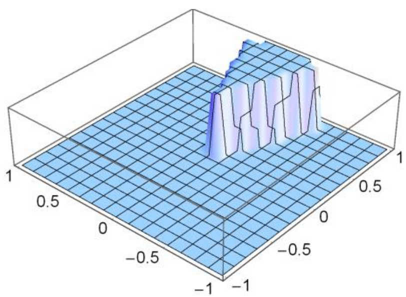


Generalized Kuwahara Filter

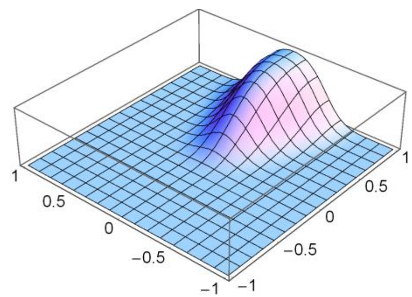
Weighting Function Construction

$$\chi_i(x,y) = \begin{cases} 1 & \dfrac{(2i-1)\pi}{N} < \arg(x,y) \le \dfrac{(2i+1)\pi}{N} \\ 0 & \text{otherweise} \end{cases} \qquad i = 1, \dots, N-1$$

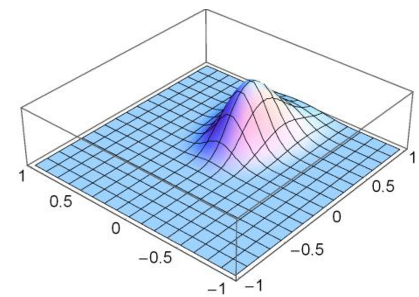
$$\begin{aligned} K_i &= (\chi_i \star G_{\sigma_s}) \cdot G_{\sigma_r} \\ &= K_0 \circ R_{-2\pi i/N} \end{aligned}$$



χ_0



$\chi_0 \star G_{\sigma_s}$



$(\chi_0 \star G_{\sigma_s}) \cdot G_{\sigma_r} =: K_0$

Generalized Kuwahara Filter

Then the mean is given by:

$$m_i = \frac{1}{|K_i|} \sum_{(x,y)\in\mathbb{Z}^2} I(x,y) \cdot K_i(x - x_0, y - y_0)$$

And the variance is given by:

$$s_i^2 = \frac{1}{|K_i|} \sum_{(x,y)\in\mathbb{Z}^2} (I(x,y) - m_i)^2 \cdot K_i(x - x_0, y - y_0)$$

The output of the generalized Kuwahara Filter is now defined by:

$$F(x_0, y_0) = \sum_{i=0}^{N-1} s_i^{-q} m_i \Big/ \sum_{i=0}^{N-1} s_i^{-q}$$

Generalized Kuwahara Filter

The parameter q is a tuning parameter that controls the sharpness of color boundaries. A typical value is $q = 8$.

$$F(x_0, y_0) = \sum_{i=0}^{N-1} s_i^{-q} m_i \bigg/ \sum_{i=0}^{N-1} s_i^{-q}$$

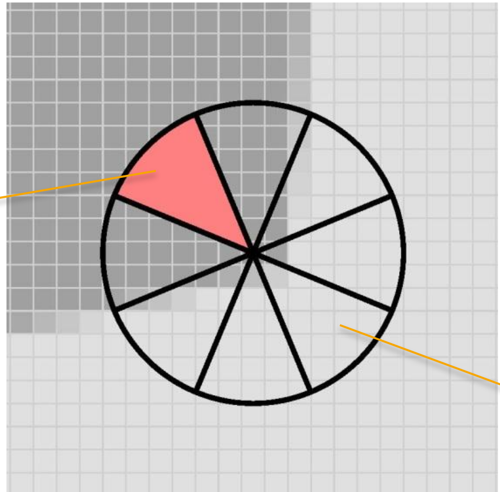
Sectors low high variance:
$s_i \to 0 \Rightarrow s_i^{-q} \to \infty$
(i.e. most influence to sum)

Sectors with high variance:
$s_i \gg 0 \Rightarrow s_i^{-q} \approx 0$
(i.e. almost no influence to sum)

Generalized Kuwahara Filter

Generalized Kuwahara filter for a corner

Sector with **small variance**. All pixels of this sector are similar. This sector contributes most to the final result

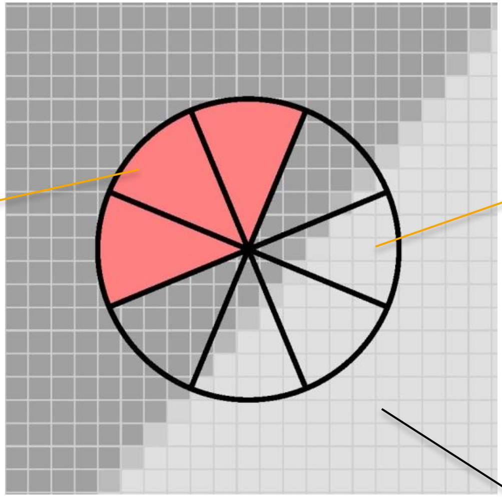


Sectors with **high variance**. They all contain pixels from both color regions. These sectors have almost no influence.

Generalized Kuwahara Filter

Generalized Kuwahara filter for an edge

Multiple sectors with **small variance**. All pixels of the sectors lie on the same side of the edge. Result is a weighted sum of the (weighted) mean values of the sectors.



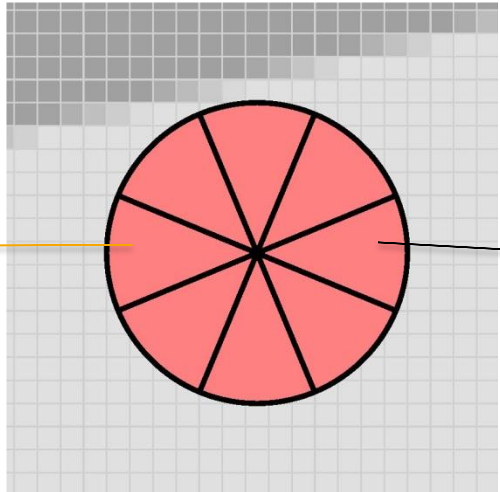
Regions with **high variance**. They all contain pixels from both sides of the edge. These sector have almost no influence.

Filter shape is similar to a truncated Gaussian

Generalized Kuwahara Filter

Generalized Kuwahara filter for an homogenous neighborhood

All regions have **similar variance**!

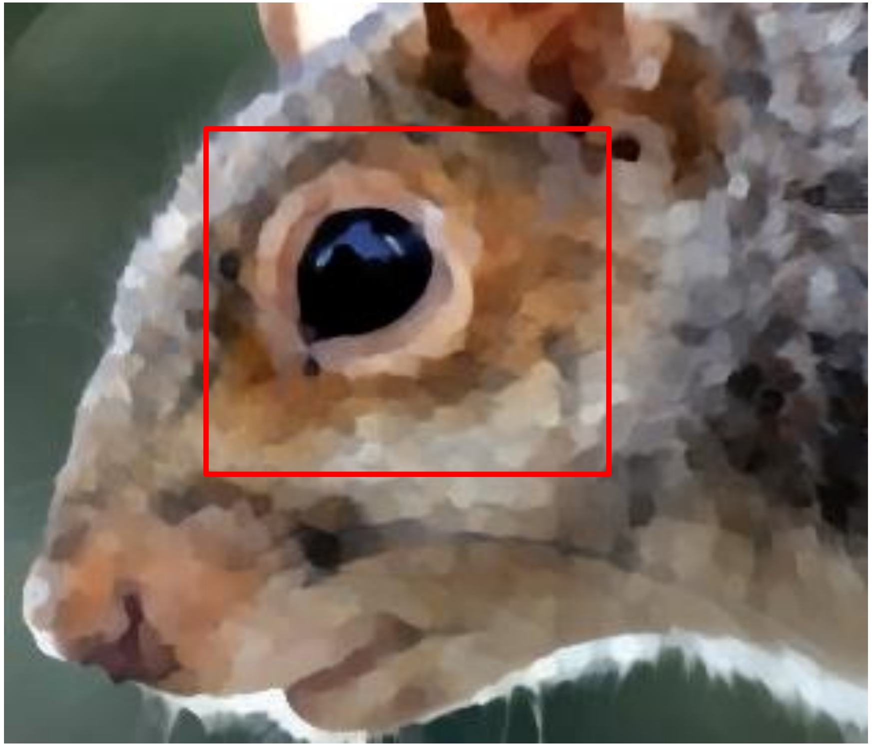


Filter shape is similar to a Gaussian filter

Generalized Kuwahara Filter

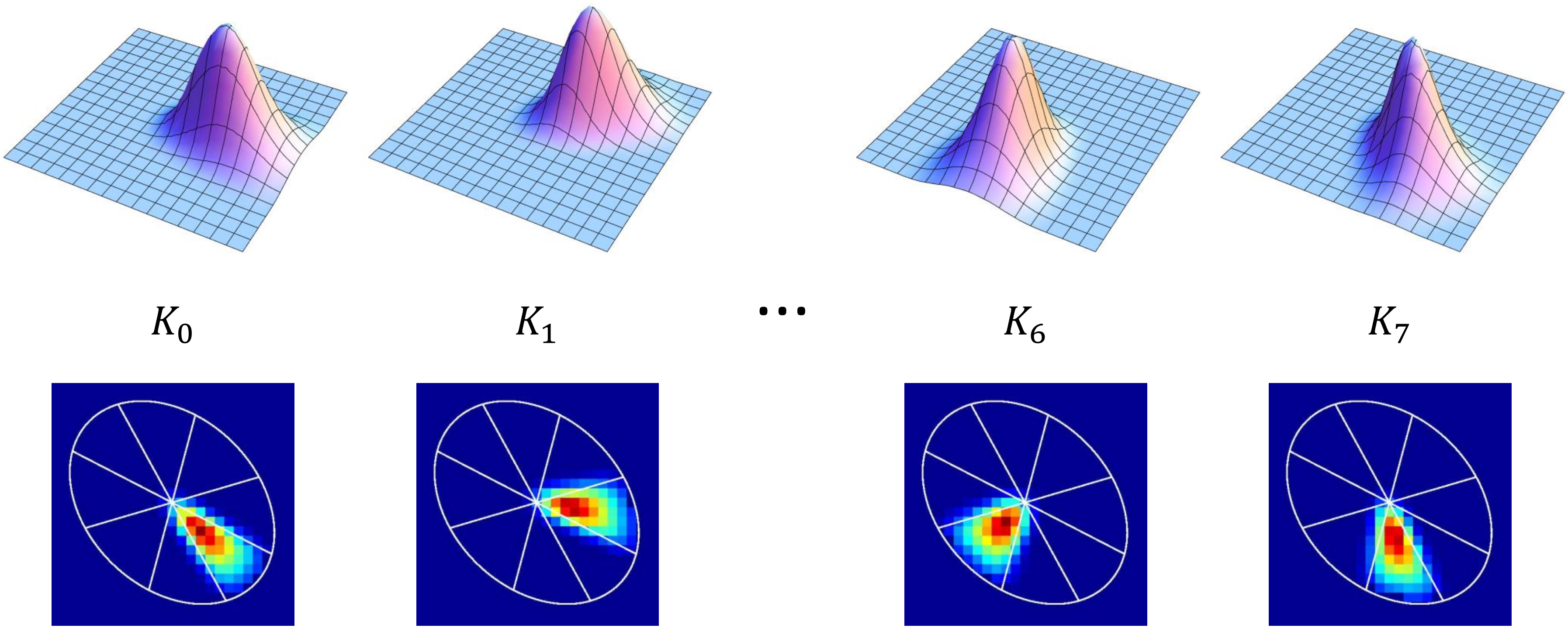
Original image by Keven Law@flickr.com

Generalized Kuwahara Filter



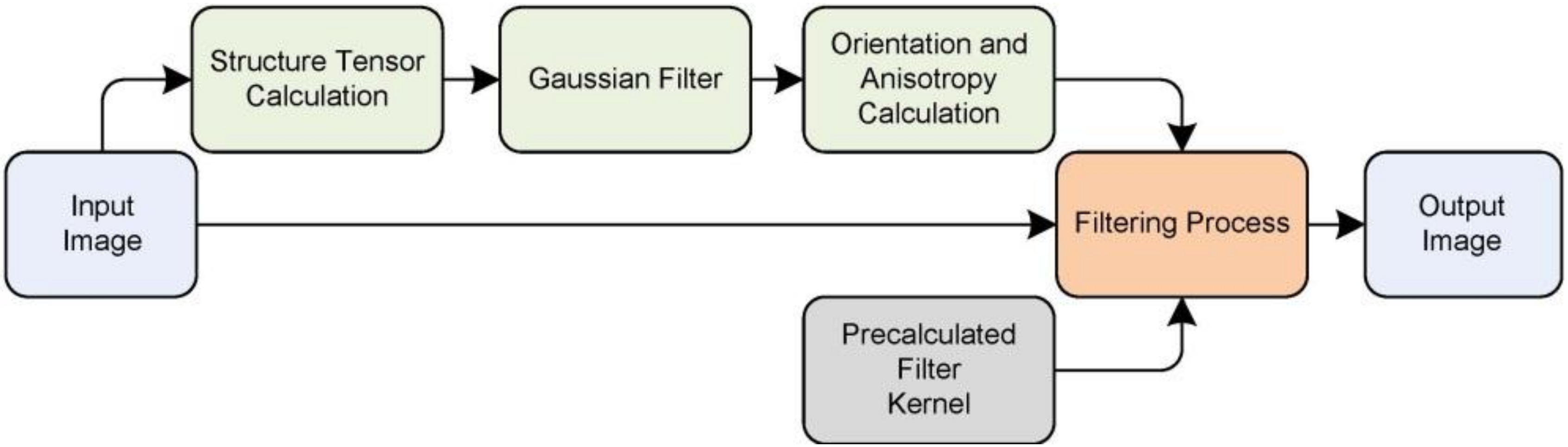
Anisotropic Kuwahara Filter

Kyprianidis et al. (2009)



Anisotropic Kuwahara Filter

Algorithm Overview

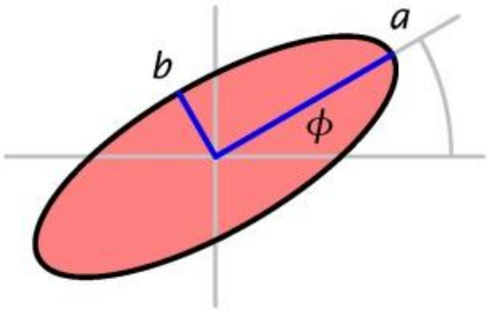


Anisotropic Kuwahara Filter

Filter Shape Definition

Elliptic filter shape (Pham, 2006)

$$a = \frac{\upsilon + A}{\upsilon} \qquad b = \frac{\upsilon}{\upsilon + A}$$



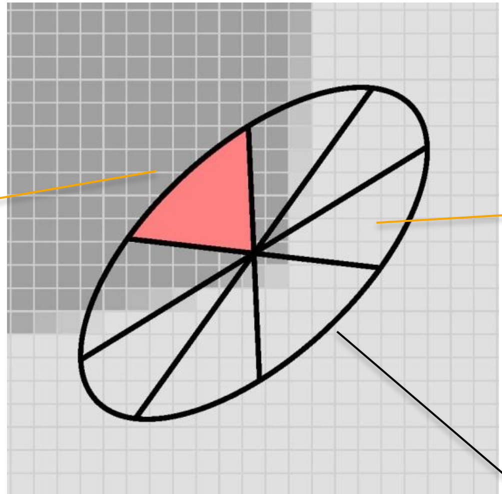
Here, $A \in [0,1]$ denotes the anisotropy measure derived from the structure tensor.

$\upsilon \in (0, \infty)$ is a user parameter that controls the eccentricity of the ellipse. A typical choice is $\upsilon = 1$.

Anisotropic Kuwahara Filter

Anisotropic Kuwahara filter for a corner

Sector with **small variance**. All pixels of this sector are similar. This sector contributes most to the final result



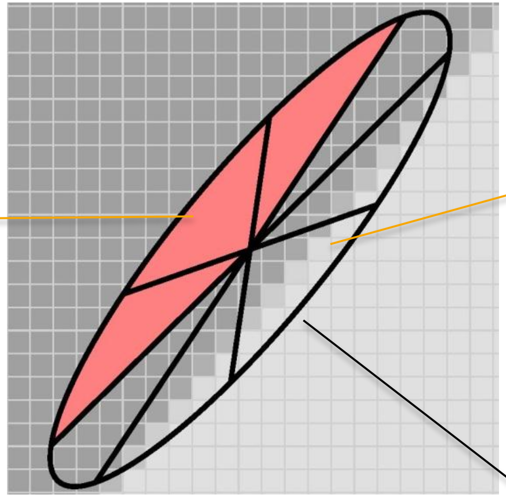
Sectors with **high variance**. They all contain pixels from both color regions. These sectors have almost no influence.

Filter shape is adapted per-pixel

Anisotropic Kuwahara Filter

Anisotropic Kuwahara filter for an edge

Multiple sectors with **small variance**. All pixels of the sectors lie on the same side of the edge. Result is a weighted sum of the (weighted) mean values of the sectors.



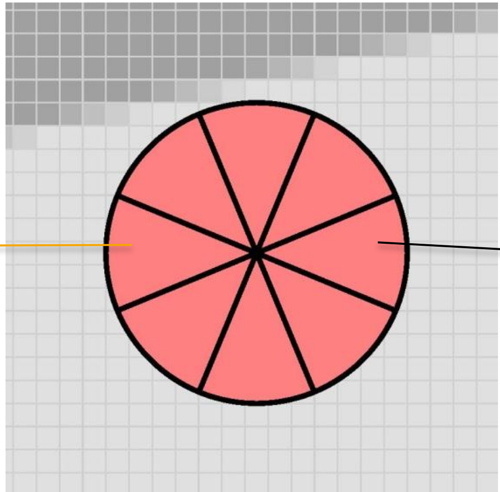
Regions with **high variance**. They all contain pixels from both sides of the edge. These sector have almost no influence.

Filter shape is adapted to anisotropic image structure

Anisotropic Kuwahara Filter

Anisotropic Kuwahara filter for an homogenous neighborhood

All regions have **similar variance**!

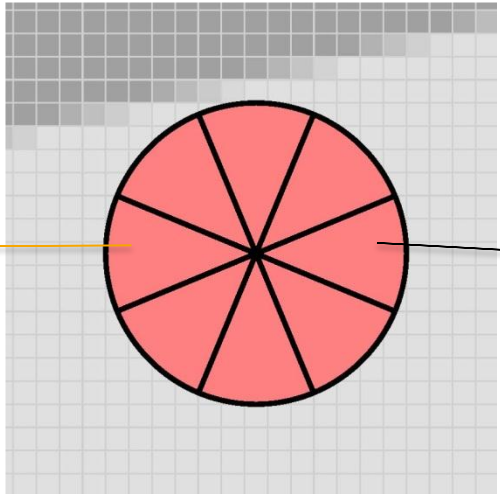


Filter shape is circular and similar to a Gaussian filter

Anisotropic Kuwahara Filter

Anisotropic Kuwahara filter for a homogenous neighborhood

All regions have **similar variance**!

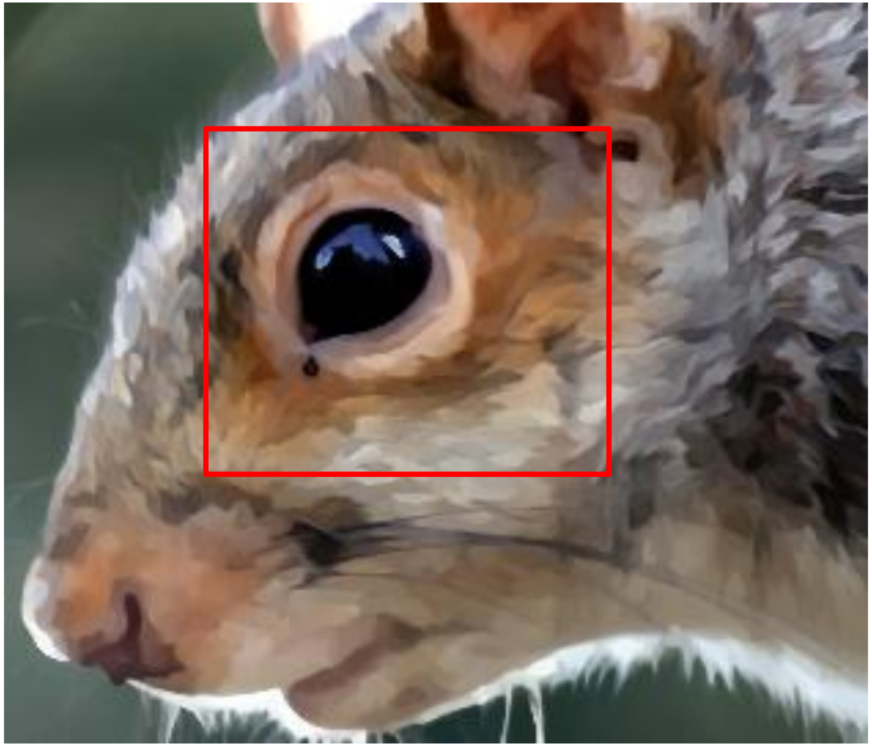


Filter shape is similar to a Gaussian filter

Anisotropic Kuwahara Filter

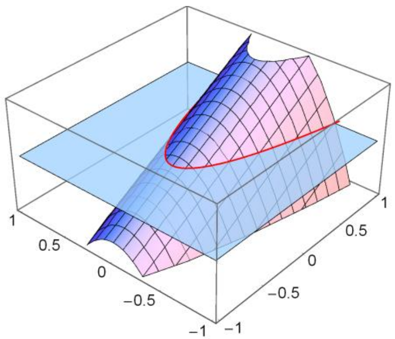
Original image by Keven Law@flickr.com

Anisotropic Kuwahara Filter

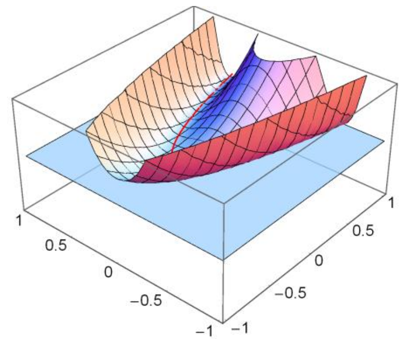


Anisotropic Kuwahara Filter

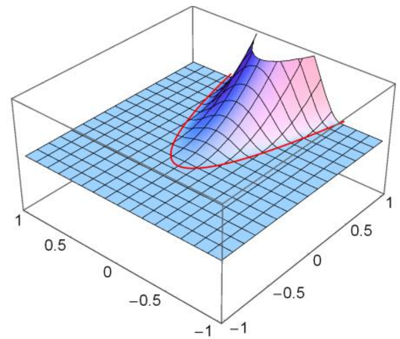
Polynomial Weighting Functions



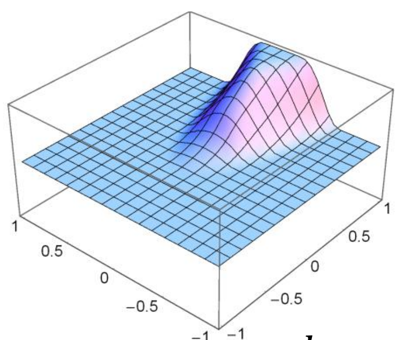
$$(x+\zeta)-\eta y^2$$



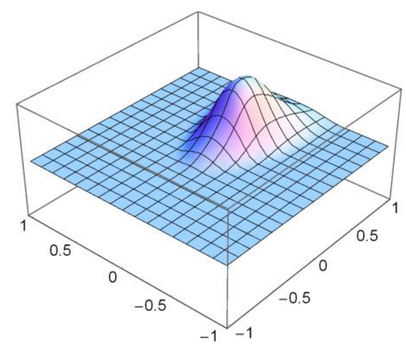
$$\left((x+\zeta)-\eta y^2\right)^2$$



$$\max\left\{0,\left((x+\zeta)-\eta y^2\right)^2\right\} \coloneqq k_0$$



$$\tilde{k}_0(x,y)=\frac{k_0}{\sum_i k_i(x,y)}$$



$$\tilde{K}_0=\tilde{k}_0\cdot G_{\sigma_s}$$

Shape-simplifying Image Abstraction

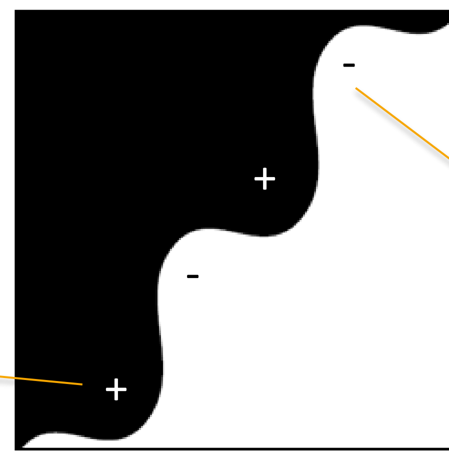
Kang & Lee (2008)

Mean curvature flow (Alvarez et al., 1992):

$$I_t = \kappa \, |\nabla I|$$

Curvature

Image gradient



Increasing the intensity here moves the level curve

Decreasing the intensity here moves the level curve

Mean curvature flow

Image credit: Kang & Lee (2008)

Input

20 iterations

60 iterations

Shape-simplifying Image Abstraction

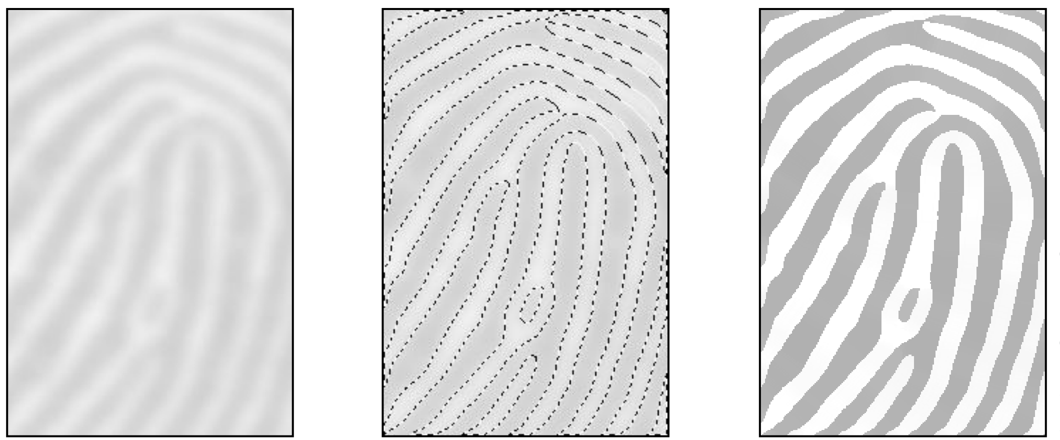
Kang & Lee (2008)

Shock filter (Osher and Rudin, 1990; Alvarez and Mazorra 1994):

$$I_t = -\mathrm{sign}(\Delta G_\sigma \star I)\ |\nabla I|$$

In the influence zone of a maximum, the Laplacian ΔI is negative and, therefore, a dilation is performed.

In the influence zone of a minimum, the Laplacian ΔI is positive, which results in an erosion.



Input | Influence zones | Output

Shape-simplifying Image Abstraction

Kang & Lee (2008)

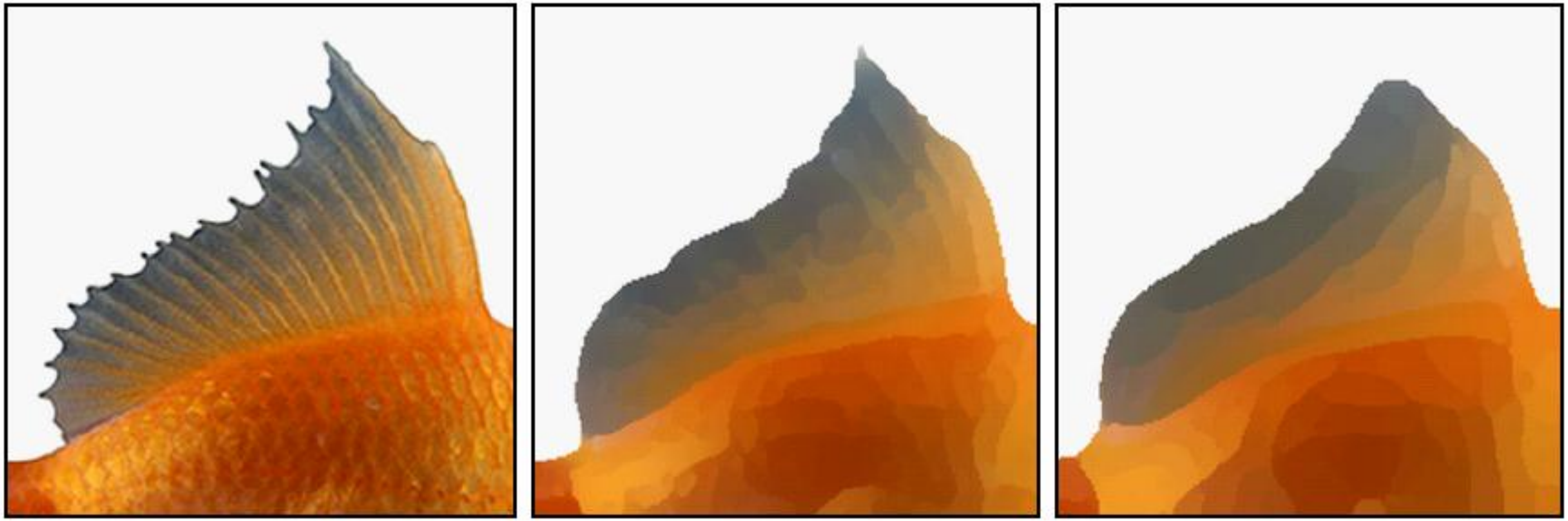
Algorithm 1 Image Abstraction by MCF

loop
  for 1 to k **do**
    $I \leftarrow MeanCurvatureFlow(I)$
  end for
  $I \leftarrow ShockFiltering(I)$
end loop

Shape-simplifying Image Abstraction
Kang & Lee (2008)

Image abstraction by mean curvature flow

Image credit: Kang & Lee (2008)



Input | 20 iterations | 60 iterations

Shape-simplifying Image Abstraction

Kang & Lee (2008)

Constrained mean curvature flow:

$$I_t = s \cdot \kappa \, |\nabla I|$$

with

$$s(x) = |t(x) \cdot \nabla I(x)|$$

Stop diffusion if grafient is not aligned to edge tangent flow (ETF)

Edge tangent flow (ETF)

Image gradient

Shape-simplifying Image Abstraction

Kang & Lee (2008)

Algorithm 2 Image Abstraction by CMCF

---

loop
  for 1 to k **do**
    $\mathbf{t} \leftarrow TVF(I)$
    $I \leftarrow ConstrainedMeanCurvatureFlow(I, \mathbf{t})$
  end for
  $I \leftarrow ShockFiltering(I)$
end loop

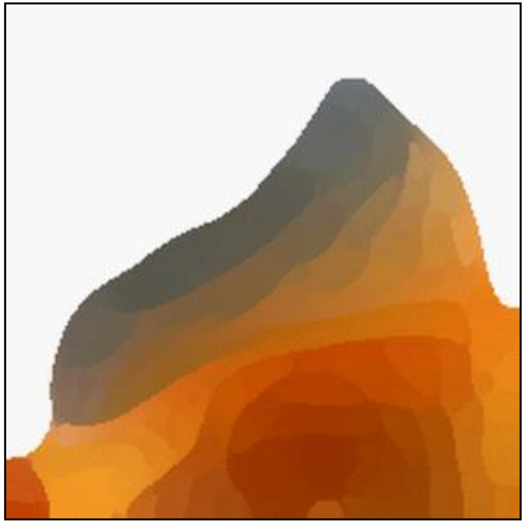
Shape-simplifying Image Abstraction

Kang & Lee (2008)

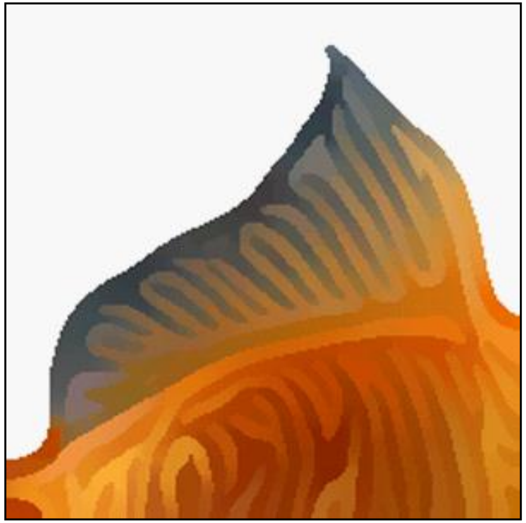
Image abstraction by constrained mean curvature flow

Image credit: Kang & Lee (2008)

Input



60 iterations image abstraction by MCF



60 iterations image abstraction by CMCF