Scale Space

Image Processing and Computer Vision

Rein van den Boomgaard   Leo Dorst

University of Amsterdam

Outline

Scale-Space

Scale-Space

We are looking for a family of image operators Ψ^s such that given the 'image at zero scale' f_0 we can construct the family of images $\Psi^s f_0$ such that:

- Ψ^s is a linear operator,
- the operator Ψ^s is translation invariant,
- it is rotational invariant,
- the operator is separable by dimension, and
- Ψ^s is invariant under changes of scale.

Scale-Space

Theorem (Gaussian Scale-Space)

The unique scale-space operator is the Gaussian convolution:

$$f(\mathbf{x}, s) = (f_0 * G^s)(\mathbf{x})$$

where:

$$G^s(\mathbf{x}) = \frac{1}{2\pi s^2} e^{-\frac{\|\mathbf{x}\|^2}{2s^2}}$$

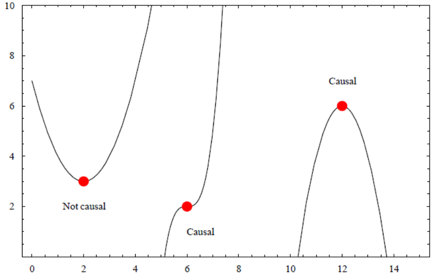
We will often write f^s to denote the image $f(\cdot, s)$.

Jan Koenderink

Prof. Jan Koenderink from Utrecht University formulated scale-space theory in the 1980's.

The structure of images
JJ Koenderink - Biological cybernetics, 1984 - Springer
Abstract. In practice the relevant details of images exist only over a restricted range of scale. Hence it is important to study the dependence of image structure on the level of resolution. It seems clear enough that visual perception treats images on several levels of resolution ... Geciteerd door 1693

Causality in Scale-Space



Luminance values (or gray values) should not be created when increasing the observation scale. I.e. a value $f(\mathbf{x}, s)$ should be traceable to a point in the infinitesimal neighborhood of $\mathbf{x}$ at an infinitesimally smaller scale $s - ds$.

Diffusion Equation

From the causality principle we can derive that the scale-space function $f(\mathbf{x}, s)$ should satisfy the partial differential equation:

$$\frac{\partial f}{\partial s} = \alpha^2 \nabla^2 f$$

with α an arbitrary function in $\mathbf{x}$ and s.

Diffusion Equation

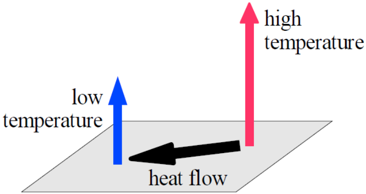
From the causality principle we can derive that the scale-space function $f(\mathbf{x}, s)$ should satisfy the partial differential equation:

$$\frac{\partial f}{\partial s} = \alpha^2 \nabla^2 f$$

with α an arbitrary function in $\mathbf{x}$ and s. Setting $\alpha^2 = s$ leads to $f_s = s\nabla^2 f$ with solution:

$$f(\mathbf{x}, s) = (f_0 * G^s)(\mathbf{x})$$

Diffusion Equation



$T_t = \nabla^2 T$ describes the temperature distribution as a function of time.

From the causality principle we can derive that the scale-space function $f(\mathbf{x}, s)$ should satisfy the partial differential equation:

$$\frac{\partial f}{\partial s} = \alpha^2 \nabla^2 f$$

with α an arbitrary function in $\mathbf{x}$ and s. Setting $\alpha^2 = s$ leads to $f_s = s\nabla^2 f$ with solution:

$$f(\mathbf{x}, s) = (f_0 * G^s)(\mathbf{x})$$

Scale-Space

Definition:

$$f(\mathbf{x}, s) = (f_0 * G^s)(\mathbf{x})$$

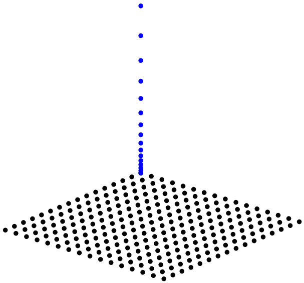
Diffusion equation:

$$f_s = s\nabla^2 f$$

Derivatives: All derivatives satisfy the diffusion equation as well:

$$(\partial_{\ldots} f)_s = s\nabla^2(\partial_{\ldots} f)$$

Sampling Scale-Space

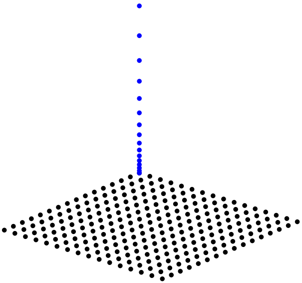


- Space is sampled *equidistantly*:

$$x = i\Delta x,\ \ y = j\Delta y$$

for $i = 0, \ldots, n_x - 1$ and $j = 0, \ldots, n_y - 1$.

Sampling Scale-Space



- Space is sampled *equidistantly*:

$$x = i\Delta x,\ \ y = j\Delta y$$

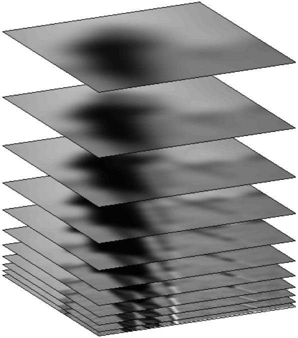
for $i = 0, \ldots, n_x - 1$ and $j = 0, \ldots, n_y - 1$.

- Scale is sampled *logaritmically*:

$$s = \alpha^i s_0,$$

for $i = 0, \ldots, n_s - 1$. (thus $\log s$ is sampled equidistantly).

Sampling Scale-Space



- Space is sampled *equidistantly*:

$$x = i\Delta x, \;\; y = j\Delta y$$

  for $i = 0, \ldots, n_x - 1$ and $j = 0, \ldots, n_y - 1$.

- Scale is sampled *logaritmically*:

$$s = \alpha^i s_0,$$

  for $i = 0, \ldots, n_s - 1$. (thus $\log s$ is sampled equidistantly).

Subsampling

512×512 256×256 128×128 64×64 32×32

$f_0 * G^{s_0}$ $f_0 * G^{2s_0}$ $f_0 * G^{4s_0}$ $f_0 * G^{8_0}$ $f_0 * G^{16s_0}$

Top row: subsampling without smoothing

Bottom row: subsampling with Gaussian smoothing

Subsampling

512×512 | 256×256 | 128×128 | 64×64 | 32×32

$f_0 * G^{s_0}$ | $f_0 * G^{2s_0}$ | $f_0 * G^{4s_0}$ | $f_0 * G^{8_0}$ | $f_0 * G^{16s_0}$

- Smoothing is *essential* before subsampling.

Top row: subsampling without smoothing

Bottom row: subsampling with Gaussian smoothing

Subsampling

512×512 | 256×256 | 128×128 | 64×64 | 32×32

$f_0 * G^{s_0}$ | $f_0 * G^{2s_0}$ | $f_0 * G^{4s_0}$ | $f_0 * G^{8_0}$ | $f_0 * G^{16s_0}$

- Smoothing is *essential* before subsampling.
- After smoothing you may subsample !

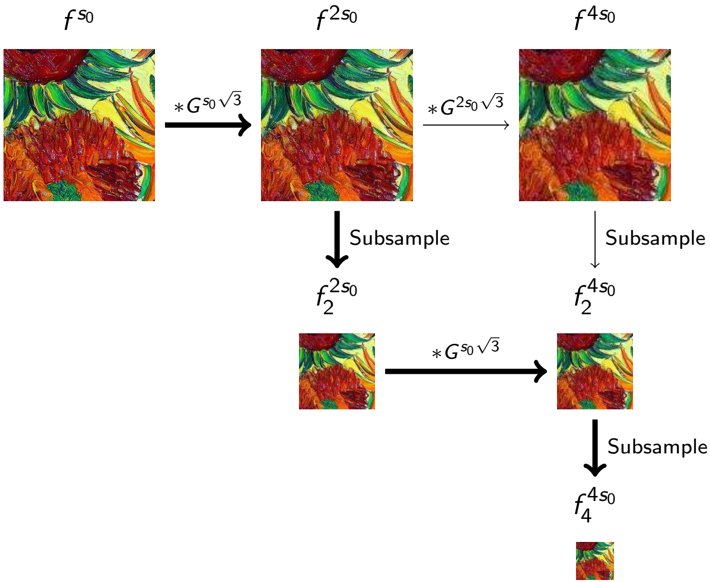
Top row: subsampling without smoothing

Bottom row: subsampling with Gaussian smoothing

Burt Image Pyramid

- Smoothing an image decreases the resolution (i.e. we are unable to resolve the small details in the image anymore).
- Effectively we are removing the high frequencies in the image.
- With enough smooting applied we may subsample the image to obtain an image with less pixels.
- If we take a Gaussian pre-smooting with scale s somewhere in the range from 0.7 to 1.5 we can reduce the number of pixels with a factor 2 in both grid directions.
- Computationally this is very attractive ...
- but this way a lot of the interesting characteristics in scale-space are missed due to the large sampling intervals in scale.

Burt Image Pyramid



Scale-Space Pyramid

Replace one stage in the Burt pyramid:

$$f^{s_0} \xrightarrow{G^{s_0\sqrt{3}}} f^{2s_0} \xrightarrow{\text{Subsample}} f_2^{2s_0}$$

with one *octave* in the scale-space pyramid:

$$f^{s_0} \xrightarrow{*G^{s_0\sqrt{\alpha^2-1}}} f^{\alpha s_0} \xrightarrow{*G^{s_0\alpha\sqrt{\alpha^2-1}}} f^{\alpha^2 s_0} \xrightarrow{*G^{s_0\alpha^2\sqrt{\alpha^2-1}}} f^{\alpha^3 s_0} = f^{2s_0} \xrightarrow{\text{Subsample}} f_2^{2s_0}$$

where $\alpha = 2^{(1/K)}$ and $K = 3$.

Scale-Space Pyramids

- Consider a scale-space sampled with $s = \alpha^i s_0$ where $\alpha = 2^{1/K}$.
- At level $i = K$ the smoothing scale is $2s_0$.
- The image at level K may thus be subsampled with a factor 2 in the spatial domain.
- In the second octave we need exactly the same Gaussian filters to obtain $f_2^{\alpha^4 s_0}$, $f_2^{\alpha^5 s_0}$ and $f_2^{\alpha^6 s_0} = f^{4 s_o}$, i.e.

$$f_2^{\alpha^4 s_0} = f_2^{2 s_0} * G^{s_0 \sqrt{\alpha^2 - 1}}$$

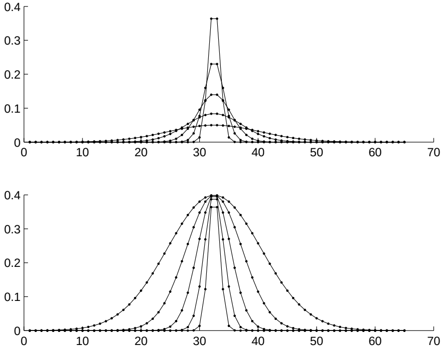
etc.

Derivatives in Scale-Space

- Scale-space: $f^s = f_0 * G^s$.
- Scale-space of derivative $g_0 = \partial f_0$ (any spatial derivative): $g^s = g_0 * G^s$.
- Differentiating and smoothing are both linear operators and so:

$$g^s = \partial f^s$$

  for any scale.
- i.e. first differentiating and then smoothing is the same as first smoothing and then differentiating.



Response of first order derivative Gaussian convolution to a step edge. On top f_x^s and on the bottom sf_x^s.

- With larger scale the response of the Gaussian first order derivative decreases.
- Plotting sf_x^s shows that a first order derivative should be multiplied with s to be constant over scale.
- *When comparing local structure (expressed in Gaussian Derivatives) across scales, it is important to use scale normalized derivatives.*

Scale Normalized Derivatives

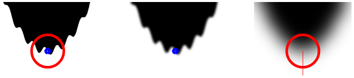
When comparing local structure (expressed in Gaussian derivatives) across all scale levels it is important to use scale normalized derivatives.

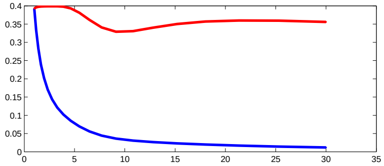
Theorem (Scale Normalized Derivatives)

Let ∂^n denote any n-th order spatial derivative then the scale normalized derivative is

$$s^n \partial^n$$

Edges in Scale-Space





- On the left the original image. The center of the circles is at $\mathbf{x}_0$.
- The middle image is the Gaussian blur at scale 2.77, the right image is at scale 21.35.
- The bottom line in the graph is the response $f_w^s(\mathbf{x}_0)$ as a function of the scale s. The top line is $sf_w^s(\mathbf{x}_0)$ plotted as a function of s.
- The two peaks in the top graph correspond with the circles drawn in the images.

Blobs in Scale-Space

Consider a Gaussian shaped blob, light on dark background:

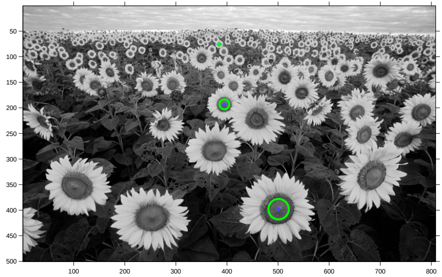
$$f_0(\mathbf{x}) = G^b(\mathbf{x} - \mathbf{x}_0)$$

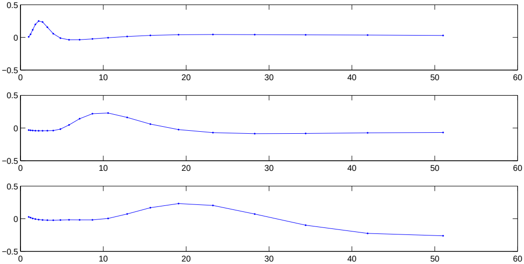
The scale normalized Laplacian operator:

$$\ell(\mathbf{x}, s) = s^2(f_0 * \nabla^2 G^s)(\mathbf{x})$$

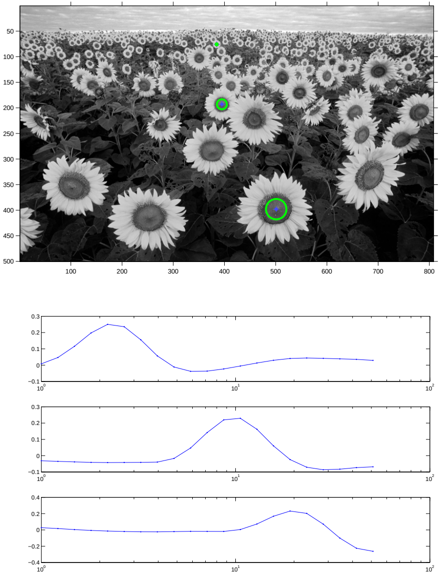
has an extremum at $s = b$ and $\mathbf{x} = \mathbf{x}_0$.

Blobs in Scale-Space





Blobs in Scale-Space



Subpixel Accurate Blob Localization

Blob detection in scale space:

- extrema of $\ell(\mathbf{x}, s) = s^2(\nabla^2 f^s)(\mathbf{x})$
- write $\mathbf{X} = \begin{pmatrix} x & y & s \end{pmatrix}^\mathsf{T}$, then $L(\mathbf{X}) = \ell(\mathbf{x}, s)$
- nescessary condition for extremum: $\nabla L = 0$
- Taylor series of L:

$$L(\mathbf{X}_0 + \mathbf{X}) = L(\mathbf{X}_0) + \mathbf{X}^\mathsf{T}(\nabla L)(\mathbf{X}_0) + \frac{1}{2}\mathbf{X}^\mathsf{T} H_L(\mathbf{X}_0)\mathbf{X}$$

- $\nabla L = 0$ then gives as subpixel accurate estimate of the extremum:

$$\mathbf{X}_0 - H_L^{-1}(\mathbf{X}_0)(\nabla L)(\mathbf{X}_0)$$