Image Mining

Vocabulary

aliasing is an effect that causes different signals to become indistinguishable (or aliases of one another) when sampled. It also often refers to the distortion or artifact that results when a signal reconstructed from samples is different from the original continuous signal.

Image convolution

Convolution:
Let f be the image and g the kernel. the output of convolution f with g is denoted f.g.

$$(f.g)_{m,n} = \sum_{k,l} f_{m-k,n-l} g_{k,l}$$

Generally, any filter kernel with positive weights and with sum equal to one will be the image smoothing filter
Gaussian filters: the filter weights are selected according to 2D Gaussian distribution,centred at the filter center with arbitrary sigma.

$$G_\sigma = \frac{1}{2\pi\sigma^2} \exp^{-\frac{x^2+y^2}{2\sigma^2}}$$

Rule: the total size of Gaussian filter equals to six sigma
Sharpening: is to apply blurring and subtract blurred image from original image.after that the information about harp edges in images is added to the original image with some scale factor.

Edge detection I

An edge is a place of rapid change in image intensity function.Such points can be identified by considering first derivative of image intensity. Edges will correspond to local extrema of derivative.
The gradient of an image: $\nabla f = [\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}]$
Gradient direction: $\theta = \tan^{-1}(\frac{\partial f}{\partial y} / \frac{\partial f}{\partial x})$
the edge strength: $||\nabla f|| = \sqrt{\frac{\partial f}{\partial x}^2 + \frac{\partial f}{\partial y}^2}$
We could approximate partial derivatives with finite difference:

Edge detection II

$$\frac{\partial f}{\partial x} \approx \frac{f(x_{xn+1}, y) - f(x_{xn}, y)}{\Delta x}$$

which can be written as simple convolution: $\begin{bmatrix} -1 & 1 \end{bmatrix}$
Other approximations of derivative filters: Roberts - Prewitt - Sobel.
Note: If image contain noise, the real edge can disappear.The solution,is to apply Gaussian smooths first to reduce the noise.

$$\frac{d}{dx}(f.g) = f.\frac{d}{dx}g$$

Canny edge detection algorithm:

- Apply Gaussian filter to smooth the image in order to remove the noise
- Find the intensity gradients of the image
- Apply non-maximum suppression to get rid of spurious response to edge detection
- Apply double threshold to determine potential edges
- Track edge by hysteresis: Finalize the detection of edges by suppressing all the other edges that are weak and not connected to strong edges.

The choice of σ depends on the desired behavior:

- Large σ detects large scale edges.
- Small σ detects fine features in images.

Formulas

number of gray levels $= N_{max} - N_{min}$
dynamics $= log_2(N_{max} - N_{min})$
Quantization $I_{quant} = \frac{I}{n_{quant}}$
the distribution of gray levels $E = \sum_{i \leqslant N} -p_i \log_2(p_i)$
Where N is the number of gray levels,p_i is the ratio (]0,1[) of pixels with gray level equal to i. This quantities measures the average number of bits per pixel necessary to encode the whole information.

Image Models I

The linear model

$$f(I+J) = f(I) + f(J)$$
$$f(\lambda I) = \lambda f(I)$$

Convolution :
Let I be a digital image. Let h be a real-valued function from $[x_1, x_2]$ x $[y_1, y_2]$.

$$(I * h)[x, y] = \sum_{i=x}^{x_2} \sum_{j=y_1}^{y_2} h[i, j] \cdot I[x-i, y-j]$$

Properties of convolution: commutativity, associativity, distributivity.

Frequency-based model

Fourier Transform
Direct

$$F[u, v] = \sum_{x=0}^{w-1} \sum_{y=0}^{h-1} f[x, y] \mathrm{e}^{-2j\pi(ux/w + vy/h)}$$

Inverse

$$f[x, y] = \frac{1}{wh} \sum_{u=0}^{w-1} \sum_{v=0}^{h-1} F[u, v] \mathrm{e}^{2j\pi(ux/w + vy/h)}$$

Properties

$$F[u, v] = \|F[u, v]\| \mathrm{e}^{j\varphi[u,v]}$$
$$F[u, v] = F[u+w, v+h]$$
$$F[u, v] = \overline{F[-u, -v]}$$
$$\|F[u, v]\| = \|F[-u, -v]\|$$
$$\varphi[u, v] = -\varphi[-u, -v]$$

$$f_1[x, y] * f_2[x, y] \to I_1[u, v] \cdot F_2[u, v]$$
$$f_1[x, y] \cdot f_2[x, y] \to F_1[u, v] * F_2[u, v]$$

$$\frac{\partial f[x, y]}{\partial x} \to \mathrm{juF}[u, v] and \frac{\partial f[x, y]}{\partial y} \to \mathrm{jvF}[u, v]$$

$$a \cdot f_1[x, y] + b \cdot f_2[x, y] \to a \cdot F_1[u, v] + b \cdot F_2[u, v]$$

Image Models II

Fourier Transform

$$f[x-x', y-y'] \to \quad F[u,v]\cdot \mathrm{e}^{-2j\pi\left(ux'/w+vy'/h\right)}$$
$$f[x,y]\cdot \mathrm{e}^{2j\pi\left(u^* x/v+v'y/h\right)} \to F[u-u', v-v']$$

Parseval theorem

$$\sum_{x=0}^{w-1}\sum_{y=0}^{h-1}\|f[x,y]\|^2 = \frac{1}{wh}\sum_{u=0}^{w-1}\sum_{v=0}^{h-1}\|F[u,v]\|^2$$

Smoothing in the Fourier domain

Low-pass Filter: The (ideal) low-pass filter is the multiplication in the frequency space by a gate function (indicator function of the 2d interval: [-u_{max} ,u_{max}] x [-v_{max} ,v_{max}])
Band-stop Filter:The band-stop filter is the multiplication in the frequency domain by a complementary band function, indicator function of the set: ($\mathbb{R}^2\backslash$[-u_{max} ,u_{max}] x [-v_{max} ,v_{max}])$\cup$[-u_{min} ,u_{min}] x [-v_{min} ,v_{min}])
High-pass Filter: The high-pass filter is the multiplication in the frequency domain by the complementary of a gate function.
Band-pass Filter: The band-pass filter is the multiplication in the frequency domain by a symmetric band function.

The statistical model

The co-occurrence matrix M_v associated to the shift vector v, is the N x N matrix (N is the number of gray levels), such that M_v(i,j) is the frequency of occurrence of couple (i,j) amongst value couples of pixels (x,x+v).
The histoglam
The histogram shows the repartition of pixels according to they (gray level) value. It provides diverse information like order statistics, entropy, and can be used in some specific cases to isolate objects.
H(x) is the amount of pixels whose gray level is equal to x.
Normalised cumulative histogram

$$HC(x) = \frac{\sum_{i=0}^{x} H(x)}{W\times H}$$

Image Models III

Histogram normalisation is an affine function of the gray level so that the image uses the whole encoding dynamics

$$f_{\text{urw}}[x,y] = (f[x,y] - \text{ Nmin })\cdot\frac{2^D-1}{\text{Nmax } - \text{ Nmin}}$$

where D: dynamics
Histogram equalisation is a function of gray levels whose purpose is to balance as uniformly as possible the gray level distribution.

$$f_{n-w}[x,y] = \left(2^D-1\right)\cdot\frac{HC(f[x,y])}{wh}$$

where (w,h): image dimensions HC(): cumulative histogram.
segmentation techniques based on grouping the gray levels from the histogram

Differential model

At the first order, to each pixel (x,y) may be associated a local frame (g,t), where the vector g corresponds to the gradient direction (i.e. line of steepest ascent), and t to the isophote direction (i.e. line of iso-gray-value).
Convolution commutes with derivation, and then estimating the derivative at a given scale is done by convolving the image by the derivative of a convolution kernel, where the spatial scope of the kernel corresponds to the scale:

$$\frac{\partial}{\partial u}(f * G_s) = f * \frac{\partial G_s}{\partial u}$$

s: estimation scale G_s : smoothing filter (scale s)
Finite differences: order 1
Approximating $\frac{\partial f}{\partial x}$ $[\mathbf{-1},\mathbf{1}]$
Approximating $\frac{\partial f}{\partial y}$ $[\mathbf{-1},\mathbf{1}]^{\intercal}$
Usually $[\mathbf{-1},\mathbf{0},\mathbf{1}]$ and $[\mathbf{-1},\mathbf{0},\mathbf{1}]^{\intercal}$ are preferred, since they produce thicker, but well centred (zero-phased)frontiers.
These operations being very noise sensitive, they are usually combined with a smoothing filter in the direction orthogonal to derivation, for example using the following kernel (or its transpose): $[\mathbf{-1},\mathbf{0},\mathbf{1}]$

Image Models IV

Finally the first order spatial derivatives in x and y may be estimated by convolving the image with the following kernels (Sobel's Masks), respectively:

$$f_x[i,j] = (f * h_x)[i,j]$$
$$f_y[i,j] = (f * h_y)[i,j]$$

$$h_x = \begin{bmatrix} -1 & 0 & 1\\ -2 & 0 & 2\\ -1 & 0 & 1\end{bmatrix} \quad h_y = \begin{bmatrix} -1 & -2 & -1\\ 0 & 0 & 0\\ 1 & 2 & 1\end{bmatrix}$$

Then to compute the norm of the gradient:

$$\|\nabla f[i,j]\|_2 = \sqrt{f_x[i,j]^2 + f_y[i,j]^2}$$
$$\|\nabla f[i,j]\|_1 = |f_x[i,j]| + |f_y[i,j]|$$
$$\|\nabla f[i,j]\|_\infty = \max\{|f_x[i,j]|, |f_y[i,j]|\}$$

and its orientation:

$$\arg(\nabla f[i,j]) = \arctan\left(\frac{f_y[i,j]}{f_x[i,j]}\right)$$

Finite differences: order 2
Approximating $\frac{\partial^2 f}{\partial^2 x}$ $[\mathbf{-1},\mathbf{-2},\mathbf{1}]$
Approximating $\frac{\partial^2 f}{\partial^2 y}$ $[\mathbf{-1},\mathbf{-2},\mathbf{1}]^{\intercal}$
The Laplacian $\Delta f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$ can then be approximated by the following linear operator(4 or 8 connected Laplacian):

$$\begin{bmatrix} & 1 & \\ 1 & -4 & 1\\ & 1 & \end{bmatrix} \quad \begin{bmatrix} 1 & 1 & 1\\ 1 & -8 & 1\\ 1 & 1 & 1\end{bmatrix}$$

Note :

$$\left.\frac{\partial f}{\partial x}\right|_{m,n} \approx \frac{1}{2\mathrm{T}_x}[f(m+1,n) - f(m-1,n)]$$

$$\left.\frac{\partial^2 f}{\partial x^2}\right|_{m,n} \approx \frac{\partial}{\partial x}\left[\frac{1}{2\mathrm{T}_x}[f(m+1,n) - f(m,n)]\right]$$

$$\left.\frac{\partial^2 f}{\partial x^2}\right|_{m,n} \approx \frac{1}{4\mathrm{T}_x^2}[(f(m+1,n) - 2\mathrm{f}(m,n)) + f(m-1,n)]$$

Erosion: $A \ominus B = \{z \mid (B)_z \subseteq A\}$
Dilation: $A \oplus B = \left\{z \mid (\hat{B})_z \cap A \neq \emptyset\right\}$

the discrete model

Tessellation of the plane A tessellation (tiling) of the plane is its partition into elementary cells (pixels). There only exist 3 regular tessellations: triangular, square, hexagonal

Every tessellation can be associated to a graph whose vertices represent elementary cells, and whose edges represent the adjacency relation between cells (2 cells are adjacent if they share an edge). Such graph is referred to as a mesh of the plane.

Note

According to the Shannon-Nyquist theorem, the sampling frequency of an image should be at least twice the highest frequency present in the image.

covariance:

The variance of a variable describes how much the values are spread. The covariance is a measure that tells the amount of dependency between two variables.

$$V(\boldsymbol{X}) = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2$$

$$\operatorname{cov}(\boldsymbol{X},\boldsymbol{Y}) = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})$$

$\sigma_{XY}^2 = 0$ iff X and Y are entirely decorrelated
$\sigma_{XY}^2 = \sigma^2$ iff A=B

basics of differential geometry

Local geometry in an image is most naturally described in terms of differential géometry: direction, curvature,...

order 1: gradient and isophote

$$\nabla I = \left(\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}\right)^T$$

Its orientation, arg ∇I, corresponds to the direction of steepest ascent.
Its magnitude, $\|\nabla I\|$, measures the local contrast.

$$\frac{\nabla I}{\nabla g} = ||\nabla I|| \text{ (main direction) ; } \frac{\nabla I}{\nabla t} = 0 \text{ (isophote)}$$

order 2: hessian and curvature

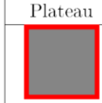
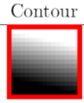
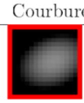
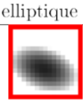
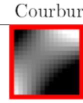
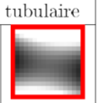
$$H_I = \begin{pmatrix} \frac{\partial^2 I}{\partial x^2} & \frac{\partial^2 I}{\partial x \partial y} \\ \frac{\partial^2 I}{\partial x \partial y} & \frac{\partial^2 I}{\partial y^2} \end{pmatrix}$$

Its eigen vectors (resp. eigen values Λ_H et λ_H) correspond to principal curvature directions (resp. intensities).
Its Frobenius norm, $\|H_I\|_F$, measures the intensity of global curvature.
Let u and v two unit vectors. The second derivative with respect to u and v is calculated as follows:

$$\frac{\partial^2 I}{\partial u \partial v} = u^T H_I v$$

representation by local derivatives

$I(x_0 + \varepsilon, y_0 + \eta) = I(x_0, y_0) + (\varepsilon, \eta)^T . \nabla I + \frac{1}{2} (\varepsilon, \eta)^T \cdot H_I \cdot (\varepsilon, \eta) + o\left(\varepsilon^2 + \eta^2\right)$
The values of derivatives up to order 2 allow dividing, depending on the dominating order, the local geometry of pixels into 4 categories.

0	1	2			
$\|\|\nabla_I\|\| \simeq 0$ $\|\|H_I\|\|_F \simeq 0$ Plateau	$\|\|\nabla_I\|\| \gg 0$ $\|\|H_I\|\|_F \simeq 0$ Contour	$\|\|H_I\|\|_F \gg 0$ $\Lambda_I \lambda_I > 0$ Courbure elliptique		$\Lambda_I \lambda_I < 0$ Courbure tubulaire	
		 $\Lambda_I < 0$ $\lambda_I < 0$	 $\Lambda_I > 0$ $\lambda_I > 0$	 $\Lambda_I < 0$ $\lambda_I > 0$	 $\Lambda_I > 0$ $\lambda_I < 0$

feature extraction and description I

Estimation of derivatives and scale spaces

$$\partial^n (I \star g) = I \star (\partial^n g)$$

In the Gaussian scale space framework, the convolution kernel g is identified to the 2d Gaussian kernel with standard deviation σ:

$$G_\sigma(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

The derivatives of image I estimated at scale σ are thus defined by the convolutions with the corresponding Gaussian derivatives:

$$\left(\frac{\partial^{i+j} I}{\partial x^i \partial y^j}\right)_\sigma = I \star \left(\frac{\partial^{i+j} G_\sigma}{\partial x^i \partial y^j}\right)$$

Corner (or Interest) points are points that carry much information relatively to the image. At the neighbourhood of these points, the image is expected to vary significantly in more than one directions.One measure of the local variations of image I at point (x,y) associated to a displacement (Δx, Δy) is the autocorrelation function:

$$\chi(x, y) = \sum_{(x_i, y_k) = W} \left(I(x_k, y_k) - I(x_k + \Delta x, y_k + \Delta y)\right)^2$$

Using a first order approximation

$$\begin{pmatrix} \Delta x & \Delta y \end{pmatrix} \begin{pmatrix} \sum\limits_{(x_k, y_k) \in W} \left(\frac{\partial I}{\partial x}(x_k, y_k)\right)^2 & \sum\limits_{(x_k, y_k) \in W} \frac{\partial I}{\partial x}(x_k, y_k) \cdot \frac{\partial I}{\partial y}(x_k, y_k) \\ \sum\limits_{(x_k, y_k) \in W} \frac{\partial I}{\partial x}(x_k, y_k) \cdot \frac{\partial I}{\partial y}(x_k, y_k) & \sum\limits_{(x_k, y_k) \in W} \left(\frac{\partial I}{\partial y}(x_k, y_k)\right)^2 \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix}$$

Autocorrelation matrix Ξ of image I at (x,y).
Corner points are those points (x,y) for which the autocorrelation matrix has two large eigen values.
The Harris detector actually calculates an interest map:

$$\Theta(x, y) = \det \Xi - \alpha \operatorname{trace}^2 \Xi$$

computing harris interest map

- Compute the first derivatives using Gaussian derivatives (standard deviation σ_1)
- Compute the components of the autocorrelation matrix $\equiv$ by using a Gaussian smoothing instead of summing on window W (standard deviation σ_2, typically $\sigma_2 = 2\sigma_1$)

feature extraction and description II

- Compute the interest map: $\Theta = \det(\Xi) - \alpha \operatorname{trace}^2(\Xi)$ (typically $\alpha = 0,06$).
- ompute the local maxima of Θ larger than a certain threshold (typically 1% of $\Theta_{\max}$).

Harris corner points obtained by calculating the first derivatives by convolution with a derivative of Gaussian of standard deviation σ.

Sift detector

The SIFT (Scale Invariant Feature Transform) detector uses The blob (elliptical structure)
For each scale-space extremum of the Laplacian representation (SIFT interest point), the associated orientation is calculated as follows:

$$\theta(x, y) = \arctan\left(\frac{G_y^\sigma(x, y)}{G_x^\sigma(x, y)}\right)$$

$$\text{with } G_x^\sigma(x, y) = \frac{\partial}{\partial x} G(x, y, \sigma) = I(x, y)^* \frac{\partial}{\partial} g_\sigma(x, y)$$

detectors and descriptors

Detector: reduce the data support (repeatable and/vs representative)

- Corners: Maxima of curvature, Harris, FAST...
- Blobs: Determinant of Hessian, SIFT, SURF...

Descriptor: data representation (invariant and/vs discriminant)

- Differential invariants: colour (intensity), contrast, Laplacian,...
- Histograms of contrast-invariant features: direction, curvature,...

Local: geometrical (contour, curvature, corner, blob...)
Global: statistical (histogram, magnitude / phase spectrum...)

Reference

References