# 1 Introduction

# 2 Image Preprocessing

## 2.1 Convolution

For two continuous functions f and g, the convolution is defined as
$\boxed{(f * g)(x) = \int_{-\infty}^{\infty} g(\tau) f(x - \tau) d\tau}$ being commutative, associative and linear.

## 2.2 Fourier Transform

The Fourier transform of $f$ is given by:
$\boxed{\hat{f}(k) := \int_{-\infty}^{\infty} f(x) e^{-2\pi ikx} dx}$ With the inverese Fourier transform:
$\boxed{f(x) := \int_{-\infty}^{\infty} \hat{f}(k) e^{2\pi ikx} dk}$

## 2.3 Impulse Functions

Dirac impulse: $\boxed{\delta(x) = \begin{cases} 0 & if x \neq 0 \\ \infty & if x = 0 \end{cases}}$ with $\boxed{\int_{-\infty}^{\infty} \delta(x) dx = 1}$

And a series of Dirac impulses: $\boxed{\sum_{n=-\infty}^{\infty} \delta(x - nT)}$

## 2.4 Nyquist Sampling Theorem

If $f$ is band bounded with cutoff frequency $k_0$ $\hat{f}(k) = 0$ for all $k$ with $|k| \geq k_0$
then it is completely determined by giving its ordinates at a series of points spaced at most $\frac{1}{2k_0}$
meaning the sample frequency must be larger than $2k_0$

## 2.5 Quantisation

Characteristic with equidistant steps of size $\Delta$

$$\boxed{g(x) = max\{0, min\{g_{max}, \left\lfloor \frac{I(x)}{\Delta} + \frac{1}{2} \right\rfloor\}\}}$$

$$\boxed{h(x) = \Delta g(x) - \frac{\Delta}{2}}$$

yielding an error of non-overdriven quantiser:

$$\boxed{I(x) - h(x) \in [-\frac{\Delta}{2}, \frac{\Delta}{2}]}$$

## 2.6 Gamma Correction

$\boxed{g_{out} = g_{max} \left( \frac{g_{in}}{g_{max}} \right)^{\gamma}}$ which keeps black and white, is a nonlinear transformation

## 2.7 Models of Blur

Blur can be modeled with convolution

$\boxed{g_{blurred} = g_{sharp} * p}$ with $p$ modeling the blur

$\boxed{p_{motion}(x) = \begin{cases} \frac{1}{n} & if -n < x \leq 0 \\ 0 & otherwise \end{cases}}$ along x-axis by $n$ pixels

$$\boxed{p_{Gauss}(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\frac{x^2}{\sigma^2}}}$$

## 2.8 Wiener Deconvoution

Restore a sharp image from a blurred image with a Wiener filter: $\boxed{g_{blurred} = g_{sharp} * p + v}$ with $p$ being a point-spread function (blur), $v$ being pixel noise and assuming $g_{sharp}$ and $v$ being independent.

With $\boxed{g_{restored} = f * g_{blurred}}$, find optimal $f$ that minimizes

$\boxed{e(k) = E\left[|\hat{g}_{sharp}(k) - \hat{g}_{restored}(k)|^2\right]}$ with $E$ being the expectation value.

## 2.9 Models of Noise

Statistical noise: $\boxed{g_{noisy}(x, y) = g_{sharp}(x, y) + v(x, y)}$

Malfunctioning sensors: $\boxed{g_{noisy}(x, y) = \begin{cases} g_{sharp}(x, y) & with\ probability\ p \\ arbitrary & otherwise \end{cases}}$

# 3 Edge-Detection

## 3.1 Finding Edges

Approximating a derivative by difference: $\boxed{\frac{\partial g}{\partial x} \approx \frac{g(x+1) - g(x-1)}{2}}$

## 3.2 Laplace Operator

The 2D analogon to the second order derivatie is the Laplace operator:
$\boxed{\bigtriangledown^2 g = \frac{\partial^2 g}{(\partial x)^2} + \frac{\partial^2 g}{(\partial y)^2}}$ which can be appriximated with
$\boxed{\bigtriangledown^2 g \approx g(x+1,y) + g(x-1,y) + g(x,y+1) + g(x,y-1) - 4g(x,y)}$
Because the second order derivative is very noisy, so combine Laplacian with Gaussian smoothing:
$\boxed{\bigtriangledown^2 (G * g) = (\bigtriangledown^2 G) * g}$ which yields
$\boxed{\bigtriangledown^2 G = \frac{x^2 + y^2 - 2\sigma^2}{\sigma^4} G(x,y)}$ called Laplacian of Gaussian (LoG) or Mexican Hat
The Laplacian of Gaussian can be approximated by calculating the Difference of Gaussian (DoG)
$\boxed{DoG(x,y) = G_{\sigma_1}(x,y) - G_{\sigma_2}(x,y)}$

## 3.3 Corner Detection

Using dissimilarity measure to find patches of maximal dissimilarity for local moves:

$$\boxed{d := \sum_{(u,v) \in rectangle} (g(u+\Delta u, v+\Delta v) - g(u,v))^2 \approx \begin{pmatrix} \Delta u \\ \Delta v \end{pmatrix}^T \begin{pmatrix} \sum(\frac{\partial g}{u})^2 & \sum \frac{\partial g}{u}\frac{\partial g}{v} \\ \sum \frac{\partial g}{u}\frac{\partial g}{v} & \sum(\frac{\partial g}{v})^2 \end{pmatrix} \begin{pmatrix} \Delta u \\ \Delta v \end{pmatrix}}$$

For special choice of coordinate system it becomes a diagonal matrix:

$$\boxed{d := \begin{pmatrix} \Delta u \\ \Delta v \end{pmatrix}^T \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} \begin{pmatrix} \Delta u \\ \Delta v \end{pmatrix} = \lambda_1 (\Delta u)^2 + \lambda_2 (\Delta v)^2 \;\; w.l.o.g. \;\; \lambda_1 \geq \lambda_2 \geq 0}$$

# 4 Curve Fitting

## 4.1 2D Geometry of Lines

Line in normal form: $\boxed{0 = \langle \vec{n}, \vec{x} \rangle + c}$ so the distance of a point $\vec{r}$ from the line: $\boxed{d = \|\vec{l} - \vec{r}\| = |\langle \vec{n}, \vec{r} \rangle + c|}$

## 4.2 Hough Transform

Every line in 2D can be represented as: $\boxed{x\ cos\phi + y\ sin\phi + c = 0}$ so there is a 2D space of all line represented by $(\phi, c)$ called Hough Space.

## 4.3 Total Least Squares

To estimate a line through a set of points we need to search the line parameters that minimise $d_i$

$\boxed{minimise \sum_{i=1}^{N} d_i^2\ subject\ to\ \langle \vec{n}, \vec{n} \rangle = 1}$ Using the Lagrange function:

$\boxed{L(\vec{n}, c, \lambda) = \sum_{i=1}^{n} d_i^2 - \lambda(\langle \vec{n}, \vec{n} \rangle - 1)}$ then zeroing the partian derivative with respect to c and then to n, we can arrive at

$\boxed{\begin{pmatrix} \alpha & \beta \\ \beta & \gamma \end{pmatrix} \vec{n} = \lambda \vec{n}}$ hence $\lambda$ is Eigenvalue and $\vec{n}$ is Eigenvector. We get two solutions of Eigenvalue Problem: $\boxed{\lambda_1 \geq \lambda_2 \geq 0}$ where $\lambda_2$ minimises distances and $\lambda_1$ maximises distances

## 4.4 Weighted Least Squares

Because of outliers introduce weights $w_i \geq 0$, one for each edge point, then solve:

$\boxed{minimise \sum_{i=1}^{N} w_i d_i^2\ subject\ to\ \langle \vec{n}, \vec{n} \rangle = 1}$. But how to choose $w_i$? Replace $d_i$ by a term that grows more slowly!

## 4.5 M-Estimators

M-Estimator means $\boxed{minimise \sum_{i=1}^{N} \rho(d_i) \ subject\ to\ \langle \vec{n}, \vec{n} \rangle = 1}$ with a suitable function $\rho$

The derivatives of the Lagrange function of M-Estimators approach and Weighted Least Squares approach are equal if $\boxed{w_i = \frac{\frac{\partial \rho(d_i)}{\partial d_i}}{2d_i}}$. So running Weighted Least Squares with appropriate weights implements M-Estimators. E.g.:

$$\boxed{\rho_{Cauchy} : d \mapsto c^2 log(1 + \frac{d^2}{c^2})}$$

$$\boxed{\rho_{Huber} : d \mapsto \begin{cases} d^2 & if|d| \leq k \\ 2k|d| - k^2 & otherwise \end{cases}}$$

## 4.6 Least Trimmed Sum of Squares Estimator (LTS)

$\boxed{minimise \sum_{i=1}^{p} d_{i:N} \ subject\ to\ \langle \vec{n}, \vec{n} \rangle = 1}$ with $p < N$ and $d_{i:N}$ the i-th element in the ordered list of point-distances. $p$ is typically given as a percentage of N.

## 4.7 RANSAC

Search a line that passes nearby as many points as possible.

$$\boxed{minimise \sum_{i=1}^{N} \sigma(d_i) \ with\ \sigma(d_i) = \begin{cases} 0 & if|d_i| \leq \theta \\ 1 & if|d_i| > \theta \end{cases}}$$

## 4.8 Estimating Circles

Parametric representation of circles:

$$\boxed{(x - m_1)^2 + (y - m_2)^2 - r^2 = 0}$$

Euclidean distance of point (x,y) from the circle:

$$\boxed{d_E = \left| \sqrt{(x - m_1)^2 + (y - m_2)^2} - r \right|}$$

Algebraic distance:

$\boxed{d_A = \left| (x - m_1)^2 + (y - m_2)^2 - r^2 \right|}$ Because minimising the Euclidean distance cannot be solved analytically, minimalize the sum of algebraic distances. Use a weighted approach, chose $w_i$ so that we implement Euclidien by using weighted Algebraic.

# 5 Color

# 6 Segmentation

## 6.1 Level Set Evolution

## 6.2 Mumford-Shah

Shrink contour if $\boxed{(I - C_{in})^2 > (I - C_{out})^2}$

Expand contour if $\boxed{(I - C_{in})^2 < (I - C_{out})^2}$

Mumford-Shah based segmentation:

$$\boxed{\frac{\partial \vec{x}}{\partial t} = \frac{\triangledown \phi}{\| \triangledown \phi\|}(C + P - \lambda_1 (I - C_{in})^2 + \lambda_2 (I - C_{out})^2)}$$