1 Image formation and acquisition

1.1 pinhole camera

simplest imaging device
geometrically, image is achieved by drawing straight lines from points in the scene through the pinhole to the image plane

1.2 Perspective projection

vars of intrest:

- M(x,y,z): scene point
- m(u,v): corresponding image point
- I: image plane
- C: optical centre
- Line through C and orth to I: optical axis
- c: intersec between optical axis and image plane (piercing point/image center)
- f: focal length
- F: focal plane

equations that relate 3d obj coords (x,y,z) to image coords:

$$\frac{u}{x} = \frac{v}{y} = -\frac{f}{z} \tag{1}$$

to get rid of up/down and left/right inversions we can write

$$u == x\frac{f}{z}; \quad v = y\frac{f}{z} \tag{2}$$

going from 3D to 2D implies a loss of information

1.3 stereo vision

Standard stereo geometry:
-Cameras arranged so that there is only horiz translation between the two CRF. Naming b the distance between the two CRF the transformation between the CRF is

$$p_L - p_R = \begin{bmatrix} b \\ 0 \\ 0 \end{bmatrix} \tag{3}$$

with p_L, p_R coords in left and right CRF
-same focal length for for both cameras (coplanar image planes)

$$x_L - x_R = b$$
$$y_L = y_R = y$$
$$z_L = z_R = z$$
$$v_L = v_R = y\frac{f}{z}(*)$$
$$u_L = x_L\frac{f}{z}$$
$$u_R = x_R\frac{f}{z}$$
$$\Rightarrow u_L - u_R = b\frac{f}{z}$$
$$u_L - u_R = d(disparity)$$
$$\Rightarrow d = b\frac{f}{z}$$
$$\Rightarrow z = b\frac{fe}{d}$$

eq. for y(*) helps solve correspondence as corresponding points have the same y coord in both images ⇒ better efficiency and lower risk of mistakes due to smaller search set

1.4 Epipolar geometry

The search space of the stereo correspondence problem is always 1D because all possible 3D points that can be correspondent to a pixel lie along a line (optical ray)
epipolar line: projection of an optical ray of one camera onto the image of the other (useful if not in standard stereo geometry)
Standard stereo geometry is more computationally efficient
Dense depth map: image in which depth is known for every pixel
images from epipolar cameras can be rectified for computational efficiency

1.5 properties of perspective projection

- the transformation is non-linear
- perspective projection maps 3D lines into image lines (straight)
- ratios of length are not preserved unless the scene is planar and parallel to the image plane
- parallelism between 3D lines is not preserved: they converge in a vanishing point unless the lines are in a plane parallel to the image plane

- the vanishing point of a 3D line is the image of the point at infinity of the line
- that is the intersection of between the parallel to the line passing through the optical center and the image plane

A scene point is in focus when all its light rays converge into a single point: always true for pinhole camera. Drawback: long exposure time is needed due to infinitesimal pinhole and scene has to be static.
Solution: lenses that gather light and focus it in a single point

1.6 thin lens model

$$\frac{1}{u} + \frac{1}{v} = \frac{1}{f} \tag{4}$$

- rays parallel to the optical axis are deflected to pass through F
- rays through C are undeflected

If the image ins in focus, the image formation process obeys the perspective projection model, with the center of the lens being the optical center and the distance v acting as the effective focal length of the projection
given an image distance, 1 scene plane is in focus
scene points outside the focusing plane are mapped into circles of confusion/blur circles
this problem is alleviated in some ways:

- method of acquisition (if blur circle is small enough it falls into one pixel)
- lens size (smaller lenses $\rightarrow$ smaller blur circles)
- diaphragm (affects effective lens size used)

F number = ratio of focal length to effective aperture (f/d) focusing mechanism: lens is moved wrt the image plane (sensor)

1.7 Image digitization

sensor converts irradiance into an electric quantity (e.g. voltage), which is sampled (2D array of pixels) and quantized (typically 8 bit in grayscale $\rightarrow$ 256 levels)
In practice image is sampled when sensed due to arrays being arranged in an array
In digital cameras sensor array=pixels
In analog cameras image from sensor array s conveerted to analog signal and then reconverted to digital but pixels $\neq$ camera sensor array
sensors are typically CCD or CMOS
CCD/CMOS is not sensitive to light wavelength $\rightarrow$ colour filtering arrays (CFA)

1.8 Camera parameters

- Signal to Noise Ratio (SNR). Main noise sources:
  - Photon Shot Noise: number of photons collected during exposure time is random (Poisson statistics)
  - Electronic Circuitry Noise
  - Quantization Noise: related to ADC conversion (in digital cameras)
  - Dark Current Noise: a random amount of charge is generated in photocapacitors in the dark due to thermal excitement. A lower bound E_{min} is set on the detected irradiation
- Dynamic Range (DR): E_{max}= irradiation that would fill up the capacity of a photodetector
  - $DR = \frac{E_{max}}{E_{min}}$
- Sensitivity: amount of signal the sensor can deliver per unit of input optical energy
- Uniformity: responses to light and dark noise vary from pixel to pixel

both SNR and DR are usually specified in decibels or bits
CCD typically provides higher SNR, DR, and better uniformity than CMOS
with CMOS electronics can be integrated on the same chip → compactness, power efficiency, lower cost.
With CMOS an arbitrary window can be read out without haveing to receive the full image
to create colour an array of CFAs is placed in front of the photodetectors so as to render each pixel sensitive to a specific range of wavelengths

1.9 Perspective projection

By appending a 1 to the Euclidean coordinates of the object we obtain the coordinates in the projective space $\mathbb{P}^3$. We assume

$$(x \quad y \quad z \quad 1) = (kx \quad ky \quad kz \quad k) \forall k \neq 0 \tag{5}$$

these are called homogeneous coordinates (aka projective coordinates) representation of the 3D point having Euclidian coordinates (x,y,z)

1.9.1 Point at infinity of a 3D line

Considering the parametric equation of a 3D line:

$$M = M_0 + \lambda D = \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} + \lambda \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} x_0 + \lambda a \\ y_0 + \lambda b \\ z_0 + \lambda c \end{bmatrix} \tag{6}$$

in projective coordinates

$$\tilde{M} = \begin{bmatrix} M \\ 1 \end{bmatrix} = \begin{bmatrix} x_0 + \lambda a \\ y_0 + \lambda b \\ z_0 + \lambda c \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{x_0}{\lambda} + a \\ \frac{y_0}{\lambda} + b \\ \frac{z_0}{\lambda} + c \\ \frac{1}{\lambda} \end{bmatrix} \tag{7}$$

by taking the limit with $\lambda \to \infty$ we obtain the projective coordinates of the point at infinity of the given line:

$$\tilde{M}_\infty = \lim_{\lambda \to \infty} \tilde{M} = \begin{bmatrix} a \\ b \\ c \\ 0 \end{bmatrix} \tag{8}$$

- to map points at infinity from the projective space to the Euclidean space we would divide by the fourth coordinate (0)
- point (0,0,0,0) is undefined
- it can be shown that all points at infinity in $\mathbb{P}^3$ lie on a plane called plane at infinity

1.9.2 Perspective projection in projective coordinates

$$u = \frac{f}{z}x \qquad v = \frac{f}{z}y \tag{9}$$

$$M = [x, y, z]^T \qquad m = [u, v]^T \tag{10}$$

$$\tilde{M} = [x, y, z, 1]^T \qquad \tilde{m} = [u, v, 1]^T \tag{11}$$

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f\frac{x}{z} \\ f\frac{y}{z} \\ 1 \end{bmatrix} = \begin{bmatrix} fx \\ fy \\ z \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{12}$$

or in matrix notation

$$k\tilde{m} = \tilde{P}\tilde{M} \tag{13}$$

often expressed as

$$\tilde{m} \approx \tilde{P}\tilde{M} \tag{14}$$

where $\approx$ means "equal up to an arbitrary scale factor"
if we assume distances to be measured in focal lengths ($f = 1$), the PPM becomes

$$\tilde{P} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} = [I_3|0] \tag{15}$$

called the canonical PPM

1.10 Building a useful camera model

For a useful camera model we need to take into account:

- Image digitization
- rototranslation between the CRF (camera reference frame) and the WRF (world reference frame)

1.10.1 Image digitization

Digitization can be accounted for by including into the projection equations the scaling factors along the two axes due to the quantization associated with the horizontal and vertical pixel size (Δu and Δv). We also need to model the translation of the piercing point wrt the origin of the image coordinate system

$$\begin{aligned} u = \frac{f}{z}x \quad &\rightarrow u = \frac{1}{\Delta u}\frac{f}{z}x = k_u\frac{f}{z}x + u_0 \\ v = \frac{f}{z}y \quad &\rightarrow v = \frac{1}{\Delta v}\frac{f}{z}x = k_u\frac{f}{z}x + u_0 \end{aligned}$$

The PPM can therefore be rewritten as

$$\tilde{P} = \begin{bmatrix} fk_u & 0 & u_0 & 0 \\ 0 & fk_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} fk_u & 0 & u_0 \\ 0 & fk_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} = A[I|0] \quad (16)$$

Matrix A, which models the characteristics of the camera, is called *Intrinsic Parameter Matrix*
$fk_u = \alpha_u$ and $fk_v = \alpha_v$ are called horizontal and vertical scale factors

1.10.2 Rigid motion between CRF and WRF

The WRF can be related to the CRF by:

- a rotation around the optical centre (3x3 rotation matrix R)
- a translation (3x1 translation vector T)

Therefore, the relation between coords of a point in the 2 RF is:

$$W = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}, M = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \Rightarrow M = RW + T \quad (17)$$

in homogenous coordinates:

$$\tilde{W} = \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}, \tilde{M} = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \Rightarrow \tilde{M} = \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} \tilde{W} = G\tilde{W} \quad (18)$$

A 3D point is therefore mapped into the CRF as

$$k\tilde{m} = A[I|0]\tilde{M} \tag{19}$$

by considering the rigid body motion between WRF and CRF:

$$\tilde{M} = G\tilde{W} \rightarrow k\tilde{m} = A[I|0]G\tilde{W} \tag{20}$$

$$k\tilde{m} = A[I|0]\begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix}\tilde{W} \tag{21}$$

Accordingly, the general form of the PPM can be expressed as:

$$\tilde{P} = A[I|0]G \quad \text{or also} \quad \tilde{P} = A[R|T] \tag{22}$$

Matrix G, which encodes the position and orientation of the camera wrt the WRF, is called Extrinsic Parameter Matrix
To be noted: the actual DoFs of G are 6, 3 due to rotation and 3 due to translation

1.11 Homographies

1.11.1 P as a Homography

If the camera is imaging a planar scene we can assume the z axis of the WRF to be perpendicular to the plane so that all points have z coordinate 0. The PPM simplifies as follows:

$$k\tilde{m} = \tilde{P}\tilde{W} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix}\begin{bmatrix} x \\ y \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{14} \\ p_{21} & p_{22} & p_{24} \\ p_{31} & p_{32} & p_{34} \end{bmatrix}\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = H\tilde{M} \tag{23}$$

where H is denoted as a *homography*, i.e. a linear transformation between projective planes.
Akin to P, H is known up to an *arbitrary scale factor* and thus independent elements in H(3x3) are just 8.
H is always invertible
Further examples of cases where homographies are applicable:

- two images of a planar scene
- two images taken by a camera rotating about the optical centre
- two images taken by different cameras (i.e. different A) in a fixed pose (i.e. same CRF)
- two images taken by different cameras rotated about the optical centre

1.12 Lens distortion

Lens distortion is modelled through additional parameters that don't alter the form of the PPM

- Radial distortion: points at the same distance from the centre of distortion (radius) are affected in the same way by the distortion, and distortion grows with the radius. Can be split into:
  - barrel distortion
  - pincushion distortion
- tangential distortion: due to misalignment of optical components and/or defects

Distortion is modelled through a non-linear mapping between ideal image coordinates and observed (distorted) image coordinates, which is a mapping between *CONTINUOUS* coordinates

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = L(r) \begin{bmatrix} \tilde{x} \\ \tilde{y} \end{bmatrix} + \begin{bmatrix} d\tilde{x} \\ d\tilde{y} \end{bmatrix} \tag{24}$$

(x', y') are the distorted coordinates and $(\tilde{x}, \tilde{y})$ are the undistorted coordinates, r is the distance from the distortion centre (x_c, y_c) $r = \sqrt{(x - x_c)^2 + (y - y_c)^2}$
The first term regards radial distortion, while the second deals with tangential distortion
$L(r)$ is usually approximated by its Taylor series around the centre of distortion $(r = 0)$

$$L(r) = 1 + k_1 r^2 + k_2 r^4 + k_3 r^6 + \dots \tag{25}$$

$L(0) = 1$, therefore there is no distortion at the centre of distortion. The odd terms are not present beacause $L(r)$ is undefined for $r < 0$, so it can be considered to be an even function.

2 Camera calibration [INCOMPLETE]

for the sake of camera calibration, patterned surfaces such as checkboards are used to find corresponding points.

2.1 Zhang's Method

1. Acquire n images of a planar pattern with m internal corners
2. comput a homography H_i for each image
3. Refine each H_i by minimizing the reprojetion error
4. given all the H_i get an initial guess for the intrinsics A
5. Given A and H_i get an initial guess for R_i and T_i (extrinsics)

6. compute an initial guess for lens distortion k
7. refine A, R_i, T_i, k by minimizing the reprojection error

2.2 Calibration of a stereo camera

The two cameras can be separately calibrated using Zhang's algorithm. Rototranslation between the two cameras needs to be calibrated

3 Image filtering

Image filter: image processing algorithm that computes the new intensity (colour) of a pixel p based on intensities (colours) of pixels in its neighbourhood

3.1 Linear Translation-Equivariant filters

Given an input 2D signal $i(x, y)$, a 2D operator $T\{\dot{}\} : o(x, y) = T\{i(x, y)\}$ is said to be linear iff:

$$T\{ai_1(x, y) + bi_2(x, y)\} = ao_1(x, y)bo_2(x, y)$$

with $o_1(\dot{}) = T\{i_1(\dot{})\}, o_2(\dot{}) = T\{i_2(\dot{})\}$ and a,b constants.
$T\{\dot{}\}$ is said to be translation-equivalent iff:

$$T\{i(x - x_0, y - y_0)\} = o(x - x_0, y - y_0)$$

If the operator is LTE, the output signal is given by the convolution between the impulse response (point spread function), $h(x, y) = T\{\delta(x, y)\}$ of the operator and the input signal.

$$o(x, y) = T\{i(x, y)\} = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} i(\alpha, \beta)h(x - \alpha, y - \beta)d\alpha d\beta$$

LTE filters can be expressed through convolutions

3.2 Convolution

3.2.1 Properties of convolution

- Associative property
- Commutative property
- Distributivity wrt sum
- Commutation with differentiation $(f * g)' = f' * g = f * g'$

3.2.2 Correlation

The correlation of signal $i(x, y)$ wrt signal $h(x, y)$ is:

$$i(x, y) \circ h(x, y) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} i(\alpha, \beta) h(x + \alpha, y + \beta) d\alpha d\beta$$

Correlation is not commutative.
If h is an even function ($h(x, y) = h(-x, -y)$):

$$\begin{aligned} i(x, y) * h(x, y) = h(x, y) * i(x, y) &= \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} i(\alpha, \beta) h(x - \alpha, y - \beta) d\alpha d\beta \\ &= \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} i(\alpha, \beta) h(\alpha - x, \beta - y-) d\alpha d\beta \\ &= h(x, y) \circ i(x, y) \end{aligned}$$

hence if h is even, the convolution between i and h is the same as the correlation of h wrt i.

3.2.3 Discrete convolution

For discrete signals, convolution is defined as foolows:

$$O(i, j) = T\{I(i, j)\} = \sum_{m=-\infty}^{+\infty} \sum_{n=-\infty}^{+\infty} I(m, n) H(i - m, j - n) \tag{26}$$

where $I(i, j)$ and $O(i, j)$ are the discrete 2D input and output signals and $H(i, j) = T\{\delta(i, j)\}$ is the kernel of the LTE operator, i.e. the response to the 2D discrete unit impulse (Kronecker delta function) $\delta(i, j)$. Properties of convolution hold in the discrete case.

3.2.4 Practical implementation of discrete convolution

Due to the image being much larger than the kernel, commutativity is exploited in order to obtain a more efficient implementation of convolution:

$$O(i, j) = T\{I(i, j)\} = \sum_{m=-\infty}^{+\infty} \sum_{n=-\infty}^{+\infty} K(m, n) I(i - m, j - n) \tag{27}$$

This equates to sliding the kernel across the whole input image and computing the convolution at each pixel. Pixels close to the border will lack some pixels for the computation of the convolution that would be outside the image. There are two main solutions:

- CROP: common in image processing, bad with chains of convolution
- PAD: zero padding, replicate, reflect, reflect_101

Computational cost: $(2k + 1)^2$ MADs, $2(2k + 1)^2$ operations per pixel.

4 Denoising

4.1 Taking a mean across time

The output at point p is:

$$O(p) = \frac{1}{N}\sum_{k=1}^{N} I_k(p) = \frac{1}{N}\sum_{k=1}^{N}(\tilde{I}(p) + n_k(p)) = \frac{1}{N}\sum_{k=1}^{N}\tilde{I}(p) + \frac{1}{N}\sum_{k=1}^{N} n_k(p) \cong \tilde{I}(p) \tag{28}$$

where $\tilde{I}(p)$ is the ideal undisturbed signal and $n_k(p)$ is the noise by which the signal is affected. Tipycally though, only a single image is available, therefore this approach is not feasible.

4.2 Mean Filter

If we are given a single image, we may compute a mean across neighbouring pixels, i.e. a spatial mean. Mean filtering is the simplest and fastest way to denoise an image. It consists in replacing each pixel intensity with the average intensity over a chosen (square) neighbourhood. The Mean Filter is an LTE operator as it can be expressed as a convolution with a kernel, e.g. for a 3x3 mean filter:

$$\begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix} = \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

According to signal theory, the Mean Filter carries out a *low pass filtering* operation, which is referred to in image processing as *image smoothing*. Smoothing is ofthen aimed at denoising, though sometimes its purpose is to cancel out small-size unwanted details that might hinder image analysis. Mean filtering is inherently fast as multiplications are not needed. Moreover, it can be implemented very efficiently by incremental calculation schemes (Box-filtering)

4.2.1 Box-Filtering

Given boxes of size $2k+1$

$$\mu(i,j) = \frac{\sum_{m=-k}^{n=k}\sum_{n=-k}^{n=k} I(i+m, j+n)}{(2k+1)^2} = \frac{s(i,j)}{2k+1)^2}$$

Where

$$s(i, j+1) = s(i,j) + V^+(i, j+1) - V^-(i, j+1)$$

and

$$V^+(i, j+1) = V^+(i-1, j+1) + a - b$$
$$V^-(i, j+1) = V^-(i-1, j+1) + c - d$$
$$V^+(i, j+1) - V^-(i, j+1) = \Delta(i, j+1)$$

Resulting in

$$s(i, j+1) = s(i, j) + \Delta(i-1, j+1) + a - b - c + d$$

Insert image from slides
We therefore have only 5 sums per pixel, independently of kernel size.

4.3 Gaussian Filter

LTE operator whose impulse response is a 2D Gaussian function (with zero mean and constant diagonal covariance matrix). The paramenter σ sets the amount of smoothing. As σ gets larger, the amount of smoothing grows.

The discrete Gaussian kernel can be obtained by sampling the corresponding continuous function, which is however of infinite extent. A finite size must be properly chosen. As the interval $[-3\sigma, +3\sigma]$ captures 99% of the area of the Gaussian function, a typical rule of thumb dictates taking a $(2k+1) \times (2k+1)$ kernel with $k = 3\lceil\sigma\rceil$. It can be noted that, as the size of the kernel grows:

- the more accurate the discrete approximation of the ideal filter turns out
- th more the computational cost grows
- the Gaussian gets smaller and smaller as we move away from the origin

4.3.1 deploying separability

To further speed up computation, one may observe that the 2D Gaussian is the product of two 1D Gaussians, therefore the original 2D convolution may be split into the chain of two 1D convolutions along x and along y

$$I(x, y) * G(x, y) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} I(\alpha, \beta) G(x - \alpha, y - \beta) d\alpha d\beta \tag{29}$$

$$G(x, y) = G(x) G(y) \tag{30}$$

$$I(x, y) * G(x, y) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} I(\alpha, \beta) G(x - \alpha) G(y - \beta) d\alpha d\beta \tag{31}$$

$$I(x, y) * G(x, y) = \int_{-\infty}^{+\infty} G(y - \beta) \left(\int_{-\infty}^{+\infty} I(\alpha, \beta) G(x - \alpha) d\alpha \right) d\beta \tag{32}$$

$$I(x, y) * G(x, y) = (I(x, y) * G(x)) * G(y) = (I(x, y) * G(y)) * G(x) \tag{33}$$

The speed-up brought in by the separability property can be expressed as:

$$\text{2D Filter: } N_{OPS} = 2 \times (2k+1)^2 \tag{34}$$

$$\text{1D Filter: } N_{OPS} = 2 \times 2 \times (2k+1) \tag{35}$$

$$S = \frac{2 \times (2k+1)^2}{2 \times 2 \times (2k+1)} = k + \frac{1}{2} \tag{36}$$

4.4 Median Filter

A non-linear filter whereby each pixel intensity is replaced by the mediano over a given neighbourhood. Median filtering counteracts impulse noise effectively, as outliers tend to fall at either the top or bottom of the sorted intensities. Median filtering also tends to keep sharper edges than linear filters such as the mean or gaussian, it cannot however correct Gaussian-like noise. A standard preprocessing pipeline includes first a median filter followed by a gaussian filter.

4.5 Bilateral Filter

An advanced non-linear filter to accomplish the denoising of a Gaussian-like noise without blurring the image (aka *edge preserving smoothing*). The kernel is computed based on the distance from the considered pixel and the difference in intensity.

$$O(p) = \sum_{q \in S} H(p,q) I_q$$

where $H(p,q)$ is a weighing function of p(central pixel) and q(neighbour pixel)

$$H(p,q) = \frac{1}{W(p)} G_{\sigma_s}(d_s(p,q)) G_{\sigma_r}(d_r(I_p, I_q))$$

where G_{σ_s} is the spatial Gaussian, G_{σ_r} is the intensity Gaussian, d_s is the spatial distance, d_r is the range (intensity distance), and $W(p)$ is the Normalization Factor

$$d_s(p,q) = \|p - q\|_2 = \sqrt{(u_p - u_q)^2 + (v_p - v_q)^2}$$
$$d_r(I_p, I_q) = |I_p - I_q|$$
$$W(p) = \sum_{q \in S} G_{\sigma_s}(d_s(p,q)) G_{\sigma_r}(d_r(I_p, I_q))$$

4.6 Non-local Means Filter

A non-linear edge preserving smoothing filter. It considers similarities among neighbourhoods spread over the image

$$O(p) = \sum_{q \in I} w(p,q) I(q)$$

where

$$w(p,q) = \frac{1}{Z(p)} e^{-\frac{\|N_p - N_q\|_2^2}{h^2}}$$
$$Z(p) = \sum_{q \in I} e^{-\frac{\|N_p - N_q\|_2^2}{h^2}}$$

In practice, to reduce the computational burden, only a region of the image is considered rather than the whole image.