# Computer Vision : Summary

Marius

October 2021

# Edge shape detection

## RANSAC

- Select a few points
- Least square regression
- Check wether the other points fit the regression

## Hough Transform

Parameter space : (ordinate at origin, angle) Intensity of a point proportional to the nummber of points it explains.

## Fourier Transform

The image is a signal, thus a sum of sine waves.
Technically two images (frequence and phase) but we often omit the latter.
The pixel $(f, \theta)$ has an intensity proportionnal to the coefficient of the sine wave of frequency $f$ in direction $\theta$.
The Fourier transform is always symmetrical as the left part is a reflect of the right one. (origin is the center of the image)

# Fourier Transform (II)

## Invertibility and Compatibility

The Fourier transform is invertible (no loss of information)
Fourier transform (and its invert) is commutative with convolutions.
Convolutions of two images matches element-wise product in the Fourier space

## Fast Fourier Transform

This enables an efficient algorithm ($\mathcal{O}(n \log n)$) to apply convolutions using a "divide to conquer" paradigm.

# Fourier Transform : Applications

## Low/High-Filter

Apply a filter to keep only low/high frequencies in the fourier transform.

## Debluring

- No noise : Division in Fourier Space, only if the kernel is never null.
- Noise : Wiener Filter

# Bilateral filters

## Gaussian smoothing : backdraws

Gaussian smoothing are blur filters where dependencies between pixels intensities increase with their physical proximity.
This may blur edges because two neighbours separated by an edge will tend to converge one to another.

## Bilateral filter

Bilateral filter uses a gaussian smoothing $G_s$ along with another gaussian smoothing $G_r$ using the difference in intensity as the distance.
As a result, dependencies between pixels intensities increase both with their physical proximity and their intensity proximity.
The coefficient are adjusted so that a pixel is always a barycenter of the pixels in the initial image.
The resulting image is smooth (details were removed) while preserving edges.

# Bilateral filters : Remarks

## Implementation tricks

Bilateral filter are not easily computable (cannot use FFT).
However, as Gaussian filters are low-pass, we may only compute the $G_r$ on close neighbours.

## Details enhancement

On the contrary, if we want to enhace details :

- Compute $I = BF(I) + D(I)$
- Compute $I' = BF(I) + k \times D(I)$

## Cross-Bilateral Filter

This technique may be used on a pair $(I, E)$, where $E$ has been taken with flash.
We use $E$ to determine intensities proximity and apply bilateral filter on $I$.
The resul is a denoised image without the flash.

# HDH Images

## High Dynamic Range Images

Same scene captured with different exposure shots
Each shot typically captures well only a part of the scene.
Empirically, linear mapping does not work well to combine them (low-intensities pixels prevail).

## Contrast and model

A contrast is given by an increasing function $f I \to f \circ I$
Linear contrast are exposure times changes !

# HDH Images (II)

## Intensity histogram and cumulative histogram

$H_I(i) = \#\{x | I(x) = i\}$
$C_I(i) = \#\{x | I(x) < i\}$

## Interest of $C$

As $C$ is increasing, $C_{f \circ I} = C_I \circ f$
In practice : $f = C_{target}^{-1} \circ C_I$
First, we separate large scale from details using BF.
We may then for instance homogenise the picture separately
$\rightarrow$ constant $H_{target}$

## Human perception

A *cone* is basically a filter on the spectrum.
A *metamere* is a collision in our perception.
RGB seems sufficient to match any perceptible color if we can add them to both side [encoded as subtractive matching]

## Empirical Grassman's law

Color matching appears to be linear

## A fundamental assertion

I am really not interested in white balance.

## Projective space

A *projective space* is the set of equivalence classes for the relation "being colinear" in a vector space.
Basically, those are the lines passing by 0.
The projective 3D-space can be seen as the plane $z = 1$ (each point is the projection of a line)...
... to which we add the line $\{y = 1,\ z = 0\}$ (projection of lines in the plane $z = 0$)...
...to which we add a $\infty$-point (projection of the line $\{y = 0, z = 0\}$)