Image Processing and Imaging
Image Formation

Christof Kauba
Fachbereich Computerwissenschaften
Universität Salzburg

Wintersemester 2020/21

1. Exposure & Autofocus
   - Exposure
   - Autofocus
     - Active - Phase Detection
     - Passive - Contrast Detection
2. Colour Imaging Pipeline
   - Channel Matching
   - Dark Correction
   - Defect Concealment
   - Smear Correction
   - Gain Nonuniformity Correction
   - Optics Corrections
   - Stochastic Noise Reduction
   - Exposure and White Balance Correction
   - Demosaicing
   - Interpolation
     - Interpolating CFA Generated Data

Image Formation Process

Image formation process can be structured into:

- Exposure
- Autofocus control

Physically (i.e. optically, electronically, and mechanically) influencing:

- Captured visual information coming out from the sensor
- Subsequent color image processing pipeline
  - Applies image processing operations onto the captured data

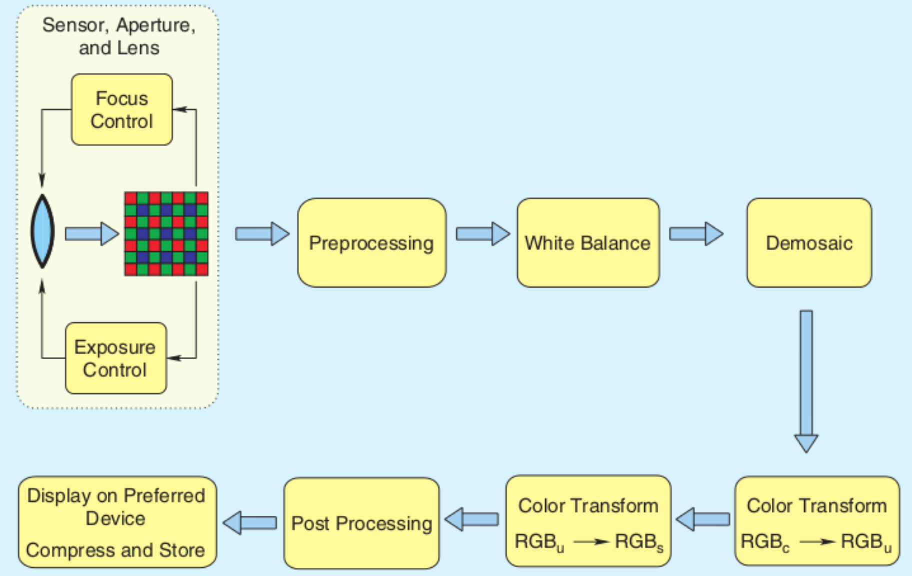


Figure: Color Imaging Pipeline: Coarse View

Outline

1. Exposure & Autofocus
   - Exposure
   - Autofocus
     - Active - Phase Detection
     - Passive - Contrast Detection

2. Colour Imaging Pipeline
   - Channel Matching
   - Dark Correction
   - Defect Concealment
   - Smear Correction
   - Gain Nonuniformity Correction
   - Optics Corrections
   - Stochastic Noise Reduction
   - Exposure and White Balance Correction
   - Demosaicing
   - Interpolation
     - Interpolating CFA Generated Data

Exposure & Autofocus

- Central aspects of image quality are contrast and sharpness
- Both aspects can be improved by image enhancement operations
- **BUT**: primarily their properties should be optimised in the image acquisition process
- This is done by exposure and focus control mechanisms in the camera

Outline

1. Exposure & Autofocus
   - Exposure
   - Autofocus
     - Active - Phase Detection
     - Passive - Contrast Detection
2. Colour Imaging Pipeline
   - Channel Matching
   - Dark Correction
   - Defect Concealment
   - Smear Correction
   - Gain Nonuniformity Correction
   - Optics Corrections
   - Stochastic Noise Reduction
   - Exposure and White Balance Correction
   - Demosaicing
   - Interpolation
     - Interpolating CFA Generated Data

Exposure (1)

Exposure is controlled by the “exposure triangle” where each item controls exposure differently:

- Shutter speed (controls the duration of the exposure)
- Aperture (controls the area over which light can enter the camera)
- ISO speed (controls the sensitivity of the camera’s sensor)

while scene luminance defines the required exposure.

- Many combinations of the above three settings lead to the same exposure.
- Key is knowing which trade-offs to make, since each setting also influences other image properties:
  - Aperture affects depth of field
  - Shutter speed affects motion blur
  - ISO speed affects image noise

Exposure (2)

- Shutter speed can be controlled with a mechanical shutter or electronically
- Aperture is controlled by the camera's iris
- ISO speed is adjusted by:
  - Either varying the amplification applied to the sensor's analog output signal before analog-to-digital (A/D) conversion
  - Or by remapping e.g. 12 bits worth of sensor CCD output onto 8 bits of digital output in the camera's A/D converter
  - In any case, noise is being amplified and even added by the first strategy

Exposure control usually requires characterisation of the brightness (or intensity) of the image:

- An over- or underexposed image will greatly affect output colors
- Depending on the measured energy in the sensor, the exposure control system changes the settings in the exposure triangle
- Both the exposure and focus controls may be based on:
  - Either the actual luminance component derived from the complete RGB image
  - Or simply the green channel data, which is a good estimate of the luminance signal

Determining the Brightness (1)

- Image is usually divided into blocks and the average luminance signal is measured in each one of these blocks
- Most common method is a centre-weighted average metering:
  - Luminance is averaged over all blocks while assigning more weight to the central 60 – 80 % of the image
- Other metering approaches:
  - Spot metering (only central image parts are being used)
  - Matrix metering (using a honeycomb configuration to identify objects and their luminance)
  - More recently, face detection is employed to identify area of specific interest (i.e. faces) to evaluate luminance and optimise exposure for such areas

- Exposure control then tries to change the exposure so that metring results fit a middle grey tone; a so called “18% grey”

Determining the Brightness (2)

Alternative to fitting the metering results to a specific average luminance value:

- Explicitly focus onto the luminance value distribution (in image regions or the entire image) by considering the image histogram.
- Histogram represents the dynamic range of the sensor and can be partitioned into e.g. 5 equally sized bins

- An underexposed image will be leaning to the left (provided that left histogram regions represent dark colours)
- An overexposed image will be leaning to the right in the histogram

- Image details disappear in over- and underexposed images
- We want as much as possible of the image to appear in the middle region of the histogram

- This can be quantified by computing the mean sample value (MSV), which determines the balance of the tonal distribution in the image (x_i ... number of pixels in histogram region):

$$MSV = \frac{\sum_{i=0}^{4}(i+1)x_i}{\sum_{i=0}^{4} x_i}$$

- Thus, the image is correctly exposed when $MSV \approx 2.5$

Determining the Brightness (3)

Distribution of luminance values as determined in the metering process can also be combined to form a measure of exposure based on the type of scene being imaged:

- Backlit
- Frontlit scene
- Nature shot

- In a typical image (nature shot), average luminance values are uniformly or randomly distributed across the scene
- Backlit or frontlit scenes may be distinguished by measuring the difference between the average luminance signal in the central area A and background area B.
- If the image is excessively frontlit, the average luminance in region A will be much higher than that in region B, and vice versa in the case of a backlit scene.

- Exposure is controlled subsequently so as to maintain the difference between the average signals in these two areas, an estimate of the object-background contrast.

Metering Problems - Over- and Under-Exposure, Dynamic Range

- All metering techniques measure the light reflected from the scene and assume all tones within the scene that they are metering to average out to a mid-grey tone
  - Might not be true for every image!
- If the scene has a lot of light tones, the camera will underexpose the image:
  - Camera's meter gives an exposure reading that renders the light tones as grey
  - This results in underexposure
- To correct this, there are exposure correction settings for taking images e.g. in the snow
- Outdoor images (and many indoor ones as well) taken with typical cameras suffer from the problem of limited dynamic range in the case of an excessively backlit or frontlit scene
- **Dynamic range** refers to the contrast ratio between the brightest pixel and the darkest pixel in an image
- Human visual system (HVS) can adapt to about four orders of magnitude in contrast ratio
- sRGB system and typical computer monitors and television sets have a dynamic range of about two orders of magnitude
- This leads to spatial detail in darker areas becoming indistinguishable from black and spatial detail in bright areas become indistinguishable from white

HDR Images

High dynamic range (HDR) solves this problem by:

(a) Capturing multiple images of the same scene at varying exposure levels on a single sensor and combining them by time multiplexing to obtain a fused image that represents the highlight (bright) and shadow (dark) regions of an image in reasonable detail

(b) Using two sensors with a different sensitivity to light avoiding temporal disturbances

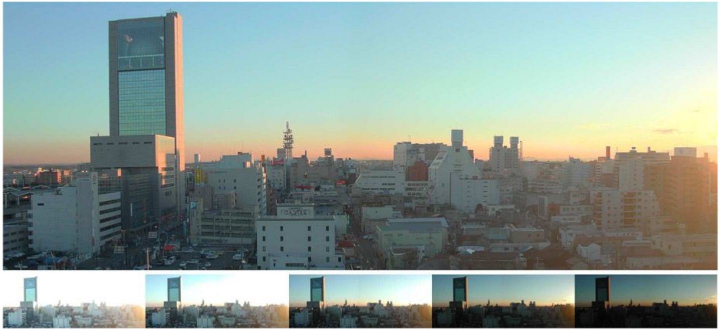


Figure: Fusing images with different exposure

Outline

1 Exposure & Autofocus
- Exposure
- Autofocus
  - Active - Phase Detection
  - Passive - Contrast Detection

2 Colour Imaging Pipeline
- Channel Matching
- Dark Correction
- Defect Concealment
- Smear Correction
- Gain Nonuniformity Correction
- Optics Corrections
- Stochastic Noise Reduction
- Exposure and White Balance Correction
- Demosaicing
- Interpolation
  - Interpolating CFA Generated Data

Autofocus - Active and Passive Systems

There are two different kinds of autofocus systems:

- *Active* AF systems:
  - Measure distance to the subject independently of the optical system
  - Subsequently adjust the optical system for correct focus
  - Various ways to measure distance, including ultrasonic sound waves (e.g. some Polaroid cameras) and infrared light (early DSC & video cameras)
- *Passive* AF systems:
  - Determine correct focus by performing passive analysis of the image that is entering the optical system
  - They generally do not direct any energy, such as ultrasonic sound or infrared light waves, toward the subject
  - However, an autofocus assist beam of usually infrared light is required when there is not enough light to take passive measurements
  - $\rightarrow$ resulting in a hybrid system
  - Passive autofocusing can be achieved by phase detection (SLR) or contrast measurement (DSC, see next slides)

Autofocus - Active and Passive Systems - Problems

- Active systems will typically not focus through windows as sound waves and infrared light are reflected by the glass
- With passive systems this will generally not be a problem, unless the window is stained
- Accuracy of active AF systems is often considerably less than that of passive systems and therefore problematic when the DoF is small
- Active systems may also fail to focus a subject that is very close to the camera since measurements get inaccurate
- As a consequence, active systems are not used in microscopy

- Passive systems may not find focus when the contrast is low
  - Notably on large single-colored surfaces (walls, blue sky, etc.) or in low-light conditions
- Passive systems are dependent on a certain degree of illumination to the subject (whether natural or otherwise)
- While active systems may focus correctly even in total darkness when necessary
- This is the motivation for the AF assist beam

See `http://graphics.stanford.edu/courses/cs178/applets/` for nice applets on this and other topics.

Autofocus - Phase Detection (1)

- Basic principle is like the split-image rangefinder focusing aid in a manual-focus SLR
- This focusing aid consists of two shallow prisms, which angle your eye's view so it sees light rays coming from the two opposite edges of the lens
- When the lens is correctly focused, these edge rays (by definition) must cross at the plane of the focusing screen
- That means objects seen by the left edge of the lens and those seen by the right edge of the lens will line up with each other as seen through the split-image prisms
- If the lens is incorrectly focused, the edge rays will cross either ahead of or behind the focusing screen
- That means the rays from the left edge and right edge will be displaced relative to each other, and lines will appear "split" through the prisms

Autofocus - Phase Detection (2)

- AF system works the same way, except that instead of the eye it uses a dedicated AF sensor consisting of two (CCD) arrays
- Optics in the AF system work the same way as the split-image prisms, directing light from the left side of the lens to one CCD, and from the right side of the lens to the other CCD (analogy to left and right eye)
- The patterns of light and dark in the subject cause the individual elements of the CCD segments to put out different values
- Total output of each CCD could be graphed as a wiggly, square-edged waveform corresponding to the light and dark patterns in the subject
- Fig. 3(a) (next slide) shows a ray diagram when the lens is in good focus and (b) shows the intensity profile corresponding to this lens position
- When the object is moved farther away, the rays from the upper and lower halves of the lens no longer intersect at the same locations
- Measured energy from the two halves of the lenses are out-of-phase (Figs. 3(c) and (d))
- Requires the lens to move relative to the image plane to compensate for this defocus; in this case, towards the image plane

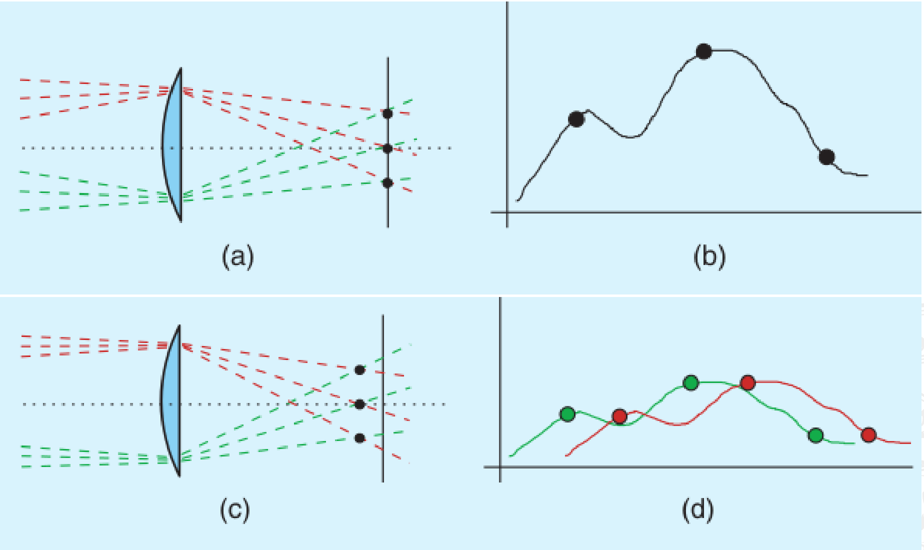


Figure: AF phase detection principle.

Autofocus - Phase Detection (4)

- AF system's CPU compares the waveforms from the two CCDs to see whether or not they are "in phase"
- If not, it can determine the amount and direction of the error based on the direction and displacement of the two waves relative to each other
- It uses this information to drive the AF motor to focus the lens
- This is why phase detection is faster compared to contrast detection, since the latter requires iterative focusing and measuring stages to determine focus.

- Fig. 4 (next slide) illustrates how this is actually done in a camera since it is not entirely obvious how to get rays from the two halves of the lens:

Autofocus - Phase Detection (5)

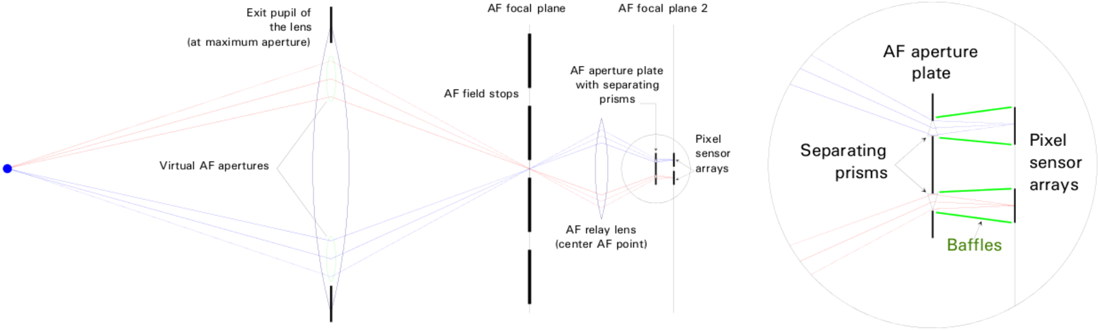


Figure: AF phase detection as used in SLR.

- NOTE: Image sensor is different from the focus sensor
  - There is a chance that they are not aligned
  - Something considered focused by the focus sensor is not always focused on the image sensor
- This is why phase-detect autofocus is more prone to front-/back-focusing issues
  - Enthusiast/high-end cameras have a micro-adjust feature to address this issue

Autofocus - Contrast Detection

- Contrast detection AF is achieved by measuring contrast (or similar values determining sharpness) within a sensor field, through the lens
- Intensity difference between adjacent pixels of the sensor naturally increases with correct image focus
- Optical system can thereby be adjusted until the maximum contrast is detected
- In this method, AF does not involve actual distance measurement at all and is generally slower than phase detection systems, especially when operating under dim light.
- As the AF system cannot calculate whether the subject is in front focus or back focus, iterative adjustment is required
- Does not use a separate sensor:
  - Contrast detection AF can be more flexible (as it is implemented in software) and potentially more accurate

Autofocus - Contrast Detection - Focusing Process

Focusing process typically consists of two components:

- An image-based measure which indicates the sharpness of the image (i.e. the degree of focus)
- A search algorithm which yields an image with the highest sharpness value
- Depending on the target hardware system, the efficiency of the search strategies is determined by:
  - Number of sharpness evaluations (i.e. the number of images being taken and evaluated)
  - Computational cost of each sharpness evaluation
  - Number of stops and directional changes of the focusing adjustment system (i.e. cost of physical lens movement)

Autofocus - Contrast Detection - Search Functions

The following search algorithms have been proposed:

- Global search: all possible focus positions are visited and the sharpness evaluated.
- Binary / Fibonacci search: A divide and conquer algorithm always breaks down the problem into two sub-problems by partitioning the focus range into sets, two equally sized sets for binary search and two sets following the golden section rule for Fibonacci search.
- Hill Climbing: the maximum is searched by going into the direction of ascending values with some larger step-size, once the values start descending, search direction is reversed and small step size is used.
- Rule-based search: depending on the value of the gradient, the focus range is partitioned into four different types of areas where the number of analysed focus position is proportional to the gradient, also descending values are recorded and are used to steer the search process. A number of rules defines how to proceed under which conditions.
- Function fitting: the position of highest sharpness is predicted from some arbitrary measurement points by fitting a function of known shape or by using a neural network.

Autofocus - Contrast Detection - Search Function Definition

- Let I denote a set of digital images which are sorted in a way that they come from a defocused state to an intermediate focus and finally to an in-focus state
- Subsequently, images get de-focused again
- An **autofocus function** is a map $f : I \rightarrow \mathbb{R}$ with the characteristic that $f(i)$ is maximised as the image comes into focus
- Further desired properties (which are eventually required to enable efficient search strategies):
  - The function should have only a single extremum, this avoids potential errors from local extrema. This means in other words that the function should be monotonically increasing towards its maximum and monotonically decreasing afterwards.
  - The extremum should be attained when the system is in focus.
  - The extremum should have a sharp peak.
  - The function should react insensitively to other parameters that possibly change during the process like the mean brightness of the image.
  - The function should be simple, thus allowing for high execution speed.

- Some focus search strategies do not require the underlying sharpness function to exhibit specific properties
- A global search is of this type, since the sharpness function does not need to obey any specific property apart from attaining its maximum when the system is in focus
- However, the number of evaluations is high when using this approach

- More efficient and intelligent schemes may significantly take advantage or even entirely rely on certain sharpness function properties to enable fast focusing
- Most of the techniques requiring a lower number of evaluations and lens movements rely on the assumption of a unimodal sharpness function
  - Binary and Fibonacci search or all variants of hill climbing

- Less stringent sharpness function properties are necessary for rule-based search (i.e. a certain extent of continuity), as well as for function fitting by using functions of known shape or by using a neural network (in the latter case it is important for the sharpness function to exhibit the same shape independent of the underlying imagery)
- In any case, also the search strategies mentioned at last take advantage of unimodality of the sharpness function since the search will terminate faster and will be more accurate

Speed is important, thus sharpness measures based on the application of initial integral transformations (like Fourier or wavelet transform) are usually not considered.
The considered spatial domain techniques can be divided into four main categories:

- **Functions based on differentiation**
- **Functions based on the histogram**
- **Functions based on statistical methods**
- **Functions based on depth of peaks and valleys**

Functions based on differentiation

- As an image comes into focus edges become sharper and therefore the amount of high spatial frequencies increases
- Image gradients are applied or the difference of the gray level intensity of pixels in the neighbourhood is calculated for computing focus measures
- Can be divided into methods that use the first derivative and methods that use the second derivative
- Examples are given in equations (1), (2), (3), and (4)

Boddeke: This method is based on applying a $(-1, 0, 1)$ filter mask along the horizontal (x) axis of an image. The focus function is defined by squaring and adding all the filtered pixel values.

$$F_{Boddeke} = \sum_{x=1}^{X-1} \sum_{y=0}^{Y} [g(x+1, y) - g(x-1, y)]^2 \ , \tag{1}$$

where X is the width of the image, Y the height of the image and $g(x, y)$ the gray level intensity of pixel (x, y).

Brenner: Brenner noted that as an image comes into focus, differences between a pixel and a pixel displaced for a certain amount increase:

$$F_{Brenner} = \sum_{x=0}^{X-n} \sum_{y=0}^{Y} [g(x, y) - g(x+n, y)]^2 \ , \tag{2}$$

where n is a number specifying the amount of displacement.

Laplace: For analysing the high frequencies of the image it is convoluted with the Laplacian operator which is a second derivative operator:

$$L = \frac{1}{4}\begin{pmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{pmatrix} .$$

The Laplace focus measure is computed as follows:

$$F_{Laplace} = \sum_{x=n}^{X-n}\sum_{y=n}^{Y-n} |L(x,y)|, \tag{3}$$

where $L(x,y)$ is the convolution of $g(x,y)$ with the mask L and n defines the size of the Laplace operator, which means that $L(x,y)$ is computed as follows:

$$L(x,y) = \frac{1}{4}\cdot[g(x,y)\cdot 4 - g(x,y+n) - g(x-n,y) - g(x,y-n) - g(x+n,y)] \quad .$$

Tenengrad: The Tenengrad autofocus function uses the Sobel operator for the calculation that in turn uses the two convolution masks

$$S_x = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} \qquad S_y = \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix} .$$

$$F_{Tenengrad} = \sum_{x=n}^{X-n} \sum_{y=n}^{Y-n} T(x, y) \;, \tag{4}$$

where $T(x, y) = S_x^2(x, y) + S_y^2(x, y)$ and $S_x(x, y)$ and $S_y(x, y)$ are the convolutions of the image with the Sobel operators S_x and S_y. Again, n determines the size of the operator.

Functions based on the histogram

- Based on the assumption that focused images have a greater number of grey levels than unfocused images
- Defocused images are expected to be a single shade of gray
- Hence the number of bins in the histogram that contain occurrences increases as the image comes into focus
- Examples are given in equations (5) and (6)

Mendelsohn and Mayall's Histogram Method: This method calculates the weighted sum of pixels in the histogram bins that are above a given threshold T and is computed as follows:

$$F_{MenMay} = \sum_{x=0}^{X} \sum_{y=0}^{Y} \begin{cases} g(x,y) \cdot H_{g(x,y)}, & g(x,y) > T \\ 0, & \text{else} \end{cases}, \tag{5}$$

where $H_{g(x,y)}$ is the number of pixels with intensity $g(x,y)$.

Range: Range is the difference between the maximum gray level and the minimum gray level, as an image comes into focus, the histogram range increases:

$$F_{Range} = max(g|H_g > 0) - min(g|H_g > 0) \ , \tag{6}$$

where H_g is the number of pixels with intensity g.

Functions based on statistical methods

- Calculate the variance or the standard deviation of the gray level intensities of an image
- Also methods that use the autocorrelation functions can be found
- This category can be divided into functions that are based on
  - Image contrast
  - Correlation measures
- Examples are given in equations (7), (8) - (10)

Variance and normalised Variance: The Variance functions are based on image contrast, which is another feature that characterises sharpness since a well-focused image can be expected to show strong variation in gray levels.

$$F_{(Nor\_)Variance} = \frac{1}{XY\overline{g}} \sum_{x=0}^{X} \sum_{y=0}^{Y} [g(x,y) - \overline{g}]^2 \;, \qquad (7)$$

where $\overline{g}$ is the mean of the gray level intensities of the image. For normalised variance, $1/\overline{g}$ is additionally used as a normalising factor to compensate for the differences in average image brightness among different images.

Vollaths' Focusing Measures: These measures are based on the autocorrelation function and the variance / standard deviation. We consider three variants:

$$F_{VollF4} = \sum_{x=0}^{X-1} \sum_{y=0}^{Y} g(x,y) \cdot g(x+1,y) - \sum_{x=0}^{X-2} \sum_{y=0}^{Y} g(x,y) \cdot g(x+2,y) \ , \tag{8}$$

$$F_{VollF5} = \sum_{x=0}^{X-1} \sum_{y=0}^{Y} g(x,y) \cdot g(x+1,y) - XY\overline{g}^2 \ , \tag{9}$$

$$F_{VollF11} = \frac{1}{XY(XY-1)} [XY \sum_{x=0}^{X-1} \sum_{y=0}^{Y} g(x,y) \cdot g(x+1,y) - (\sum_{x=0}^{X} \sum_{y=0}^{Y} g(x,y))^2] \ . \tag{10}$$

Functions based on depth of peaks and valleys

- Local extrema of the intensity values and their distances are considered
- Based on the observation that peaks and valleys are better separated in focused images
- Examples are given in equations (11) and (12)

Thresholded Content: This method adds the pixel values that are above a certain threshold T:

$$F_{Th\_Cont} = \sum_{x=0}^{X} \sum_{y=0}^{Y} \begin{cases} g(x,y), & g(x,y) \geq T \\ 0, & \text{else} \end{cases} . \tag{11}$$

Thresholded Pixelcount: This function computes the number of pixels below (above) a certain threshold:

$$F_{Th\_Pixelcount} = \sum_{x=0}^{X} \sum_{y=0}^{Y} s[g(x,y), T] \ , \tag{12}$$

with

$$s[g(x,y), T] = \begin{cases} 0, & g(x,y) \geq T \\ 1, & g(x,y) < T \end{cases} \tag{13}$$

Of course, there are also autofocus functions that combine several features of other autofocus functions.
Examples are given in equations (14, 15) and (10).

Variance of Sobel: The variance of the magnitude of the Sobel gradient is calculated.

$$F_{Var\_Sobel} = \sum_{x=n}^{X-n} \sum_{y=n}^{Y-n} (|S(x,y)| - \overline{S})^2 \ , \tag{14}$$

where $S(x,y) = \sqrt{S_x^2(x,y) + S_y^2(x,y)}$ and $\overline{S}$ is the mean of the absolute values of the Sobel gradient given by

$$\overline{S} = \frac{1}{(X-n)(Y-n)} \sum_{x=n}^{X-n} \sum_{y=n}^{Y-n} S(x,y) \ . \tag{15}$$

- Several functions depend on a threshold or can be used with specific parameters
- Their behaviour often is significantly influenced by these parameters
- Fig. 5 as an example for the Laplace function with $n = 1$ and $n = 10$, all examples computed from sequences of 40 hardness testing images with different focus

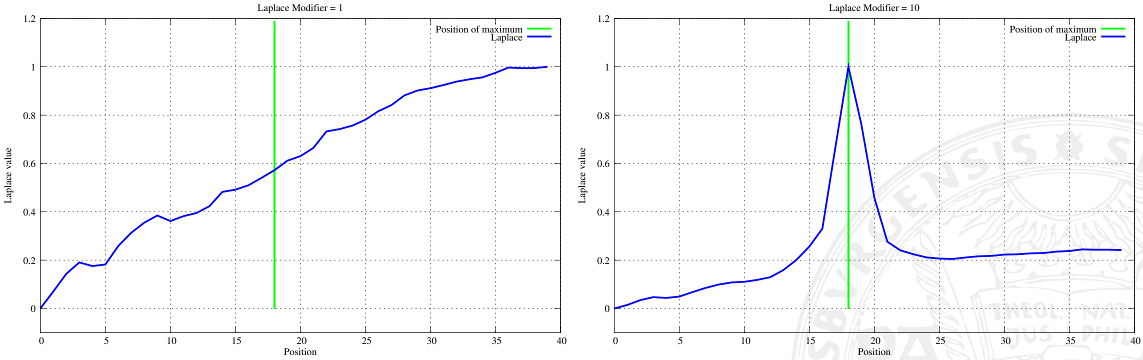
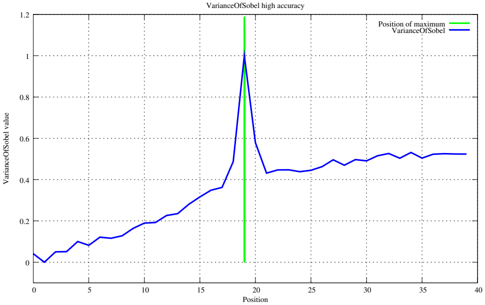


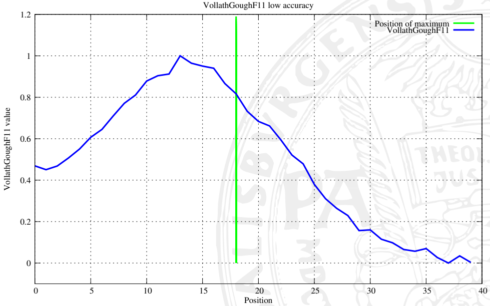
Figure: Laplace autofocus function after normalisation applied to a series of 40 images ($n = 1$, $n = 10$).

Accuracy: The most important criterion an autofocus function should fulfill is that the extremum should be attained when the image is in focus. This aspect is important for all focus search algorithms. A way to score a function for this criteria is to use:

$$F_{acc} = \frac{1}{1 + 0.25 \cdot (max_{found} - max_{true})^2},$$

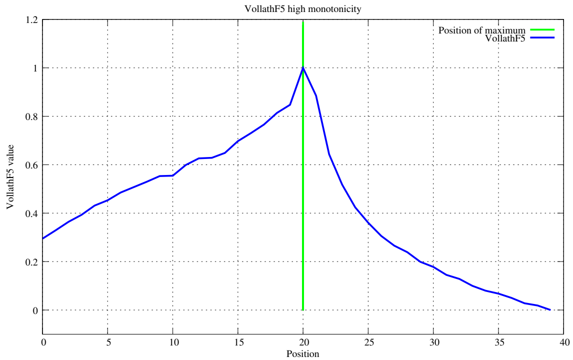
where max_{true} is the position of the sharp image in the image stack and max_{found} is the position of the image in the image series that the autofocus function has computed. $F_{acc} = 1$, when the autofocus function has computed the right position. The higher the difference between max_{true} and max_{found} is, the more F_{acc} goes towards 0.





Monotonicity:

- For all focus search algorithms relying on a unimodal sharpness function, the function should be monotonically increasing towards its maximum and monotonically decreasing afterwards
- A function F_{mon} has been used that calculates the differences of all $F(i)$ and $F(i+1)$ within the image series, where $F(i)$ denotes an autofocus functions' value of the image on position i
- $F_{mon} = 1$, when the autofocus function is monotonically increasing towards its maximum and monotonically decreasing afterwards
- The more often the monotonicity is disturbed, the more F_{mon} goes towards 0
- Therefore initially $F_{mon} = 1$, each time the monotonicity is disturbed, 0.075 is subtracted
- When the value becomes negative $F_{mon} = 0$ and the algorithm stops
- It should be noted that autofocus functions which produce more than a single extremum are scored low by F_{mon}



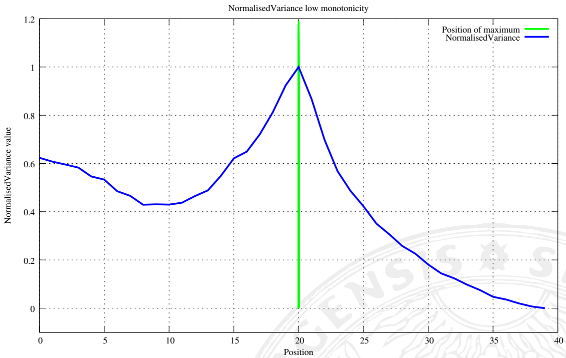
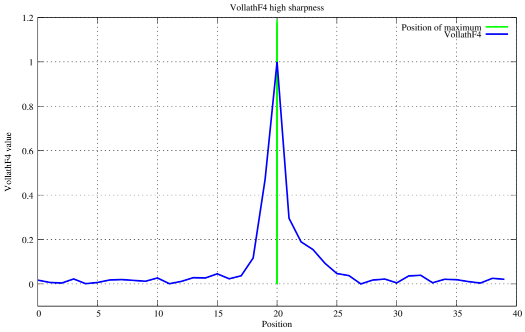


Figure: Monotonicity of autofocus function: F_{VollF5}, $F_{mon} = 1$ vs. $F_{Nor\_Variance}$, $F_{mon} = 0.325$.

Peak Sharpness:

- Sharpness of the peak is another criterion for selecting a good focus measure
- A sharp peak makes algorithms possible that do a coarse search for the peak within the calculated values and come in a finer state when the values change more significantly.
- Especially two-step search and rule-based search may significantly benefit from distinct peak sharpness

- To accomplish an assessment for that criterion a function F_{sharp} has been developed that counts the values of an autofocus function that are above a focus level of 0.3
- Few values are expected to be above that threshold if the autofocus function has a sharp peak
- Therefore $F_{sharp} = 1$, in case less than 20 percent of the values are above 0.3
- The more values are higher than 0.3, the more F_{sharp} goes towards 0
- $F_{sharp} = 0$, as soon as more than 50 percent of the values are above 0.3



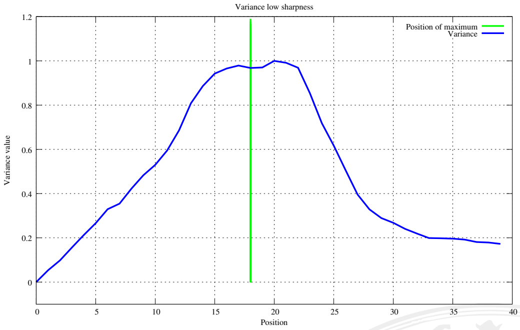


Figure: Peak Sharpness of autofocus function: F_{VollF4}, $F_{sharp} = 1$ vs. $F_{Variance}$, $F_{sharp} = 0$.

Autofocus/focus on white or other uniform areas:

- Neither passive method will focus well if there isn't a contrast change in the image
- E.g. a solid white wall (or white portions) or concrete floor or blue sky with no clouds
- Measurement is not about distance or intensity of light but contrast within the image
- This is why some assist beam systems use a red grid pattern to help AF in low-contrast situations

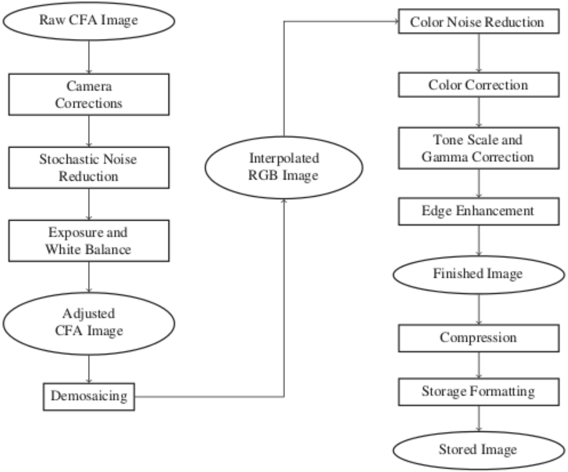
Outline

1 Exposure & Autofocus
- Exposure
- Autofocus
  - Active - Phase Detection
  - Passive - Contrast Detection

2 Colour Imaging Pipeline
- Channel Matching
- Dark Correction
- Defect Concealment
- Smear Correction
- Gain Nonuniformity Correction
- Optics Corrections
- Stochastic Noise Reduction
- Exposure and White Balance Correction
- Demosaicing
- Interpolation
  - Interpolating CFA Generated Data

- Operates on the acquired image, in most cameras based on three differently populated colour planes
- The first processing block in the pipeline depicted in Fig. 8, "camera correction", is actually a collection of blocks as detailed in Fig. 9

- Processing blocks required for a specific camera vary depending upon the hardware and the user expectations
- Lower cost hardware typically leaves more artifacts in the raw image to be corrected, but user expectations are often lower as well
- Choice of correction blocks used with a particular camera is the result of a number of system engineering and budget decisions
- Few, if any, cameras use all of the processing blocks shown in Fig. 9



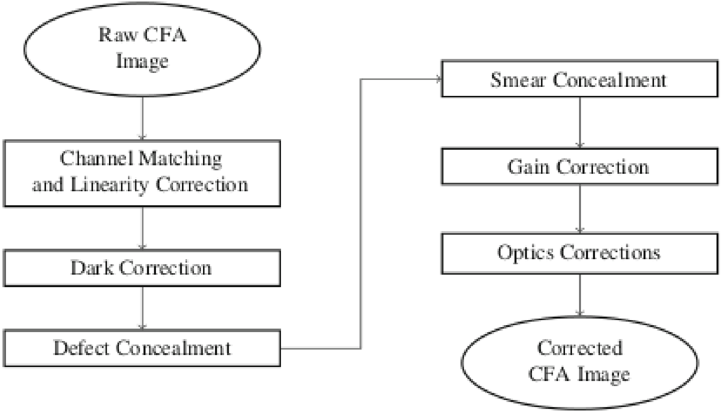


Figure: The stages of camera correction

Outline

1 Exposure & Autofocus
- Exposure
- Autofocus
  - Active - Phase Detection
  - Passive - Contrast Detection

2 Colour Imaging Pipeline
- Channel Matching
- Dark Correction
- Defect Concealment
- Smear Correction
- Gain Nonuniformity Correction
- Optics Corrections
- Stochastic Noise Reduction
- Exposure and White Balance Correction
- Demosaicing
- Interpolation
  - Interpolating CFA Generated Data

Channel Matching

The first correction is to match the response of multiple outputs or analog signal processing chains, such as with a dual output sensor:

- Artifacts due to channel mismatch are highly structured
- Usually a seam in the middle of the image or a periodic column pattern, the responses for the multiple outputs must match very closely

- The most common form of this correction:
- Adaptively compute a dark offset correction for each output
- That will bring similar pixels from each output to match a common value, using reference dark pixels

- The key to successful matching of multiple output channels is to take advantage of the knowledge of which image pixels came from which output

Outline

1. Exposure & Autofocus
   - Exposure
   - Autofocus
     - Active - Phase Detection
     - Passive - Contrast Detection
2. Colour Imaging Pipeline
   - Channel Matching
   - Dark Correction
   - Defect Concealment
   - Smear Correction
   - Gain Nonuniformity Correction
   - Optics Corrections
   - Stochastic Noise Reduction
   - Exposure and White Balance Correction
   - Demosaicing
   - Interpolation
     - Interpolating CFA Generated Data

Dark Correction

- Dark correction is always necessary:
- Analog output from the image sensor is rarely precisely “zero” for a zero light condition
- Even with the lens cap on, a dark current signal is recorded, which is due to thermally generated electrons in the sensor substrate
- To account for this, two strategies are used:
  - Place an opaque mask along the edges of the sensor to give an estimate of intensity due to dark current alone (this value can be corrupted by noise in the dark pixels, so some smoothing may be used to reduce the dark floor estimation error)
  - Capture a dark image for the given exposure time (a second image is taken immediately after capturing the scene image with no exposure).
- In the first case, the mean dark current is subtracted from the entire image (this only works well in case of uniform dark floor)
- In the second one, the dark image itself is subtracted from the captured data

- In some cases, the dark floor is modeled using data from multiple dark captures
- By averaging, the impact of temporal noise on the dark floor estimate is minimised
- This technique is still affected by changes in sensor temperature and integration time
- Astronomical and other scientific applications, especially ones using temperature-controlled sensor, routinely use this technique, made easier by the controlled temperature

Outline

1. Exposure & Autofocus
   - Exposure
   - Autofocus
     - Active - Phase Detection
     - Passive - Contrast Detection
2. Colour Imaging Pipeline
   - Channel Matching
   - Dark Correction
   - Defect Concealment
   - Smear Correction
   - Gain Nonuniformity Correction
   - Optics Corrections
   - Stochastic Noise Reduction
   - Exposure and White Balance Correction
   - Demosaicing
   - Interpolation
     - Interpolating CFA Generated Data

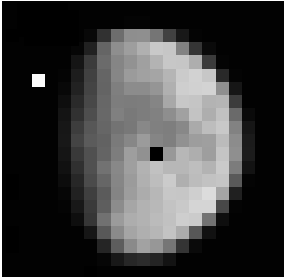
Defect Concealment (1)

- Sensor defects are somewhat problematic
- They indicate lost data that simply was not sensed
- Algorithms for treating defects interpolate the missing data
- The most common defects are isolated single pixel defects
- Concealment of isolated pixels is usually done with a linear interpolation from the nearest adjacent pixels of the same color sensitivity.
- There are two way of treating these defects:
  - Applying an impulse noise filter which tends to (inappropriately) filter out high-contrast details such as stars, lights, or specular reflections
  - Maintaining a map of defective pixels depending upon a map from the sensor or camera manufacturer

Defect Concealment (2)

- Bright pixel defects (hot pixels) caused by cosmic ray damage must be concealed without depending upon a preinstalled map
- A camera can implement a dark image capture and bright defect detection scan in firmware, usually done at startup
- New defects found in the dark image are added to the defect map
- Cosmic ray damage tends to produce bright points rather than marginal defects $\rightarrow$ detecting these defects is relatively easy

- Sensor column defects or other more clustered defects caused e.g. by dirt on the cover glass of the sensor are much more difficult to correct:
  - The latter ones also vary in size depending on the focal length of the lens
  - Amount of defective/missing data which needs to be corrected/interpolated is higher than for the single, isolated defects



Outline

1. Exposure & Autofocus
   - Exposure
   - Autofocus
     - Active - Phase Detection
     - Passive - Contrast Detection
2. Colour Imaging Pipeline
   - Channel Matching
   - Dark Correction
   - Defect Concealment
   - Smear Correction
   - Gain Nonuniformity Correction
   - Optics Corrections
   - Stochastic Noise Reduction
   - Exposure and White Balance Correction
   - Demosaicing
   - Interpolation
     - Interpolating CFA Generated Data

Smear Correction (1)

- Interline smear is a challenging artifact to correct or conceal because the artifacts vary with scene content
- It is manifested as an offset added to some of the columns in the captured image
- Since the added signal will usually vary from column to column, the effect will vary with the original scene content

- If a small amount of charge is added to pixels that are well below saturation:
- Artifact is manifested as a column that is brighter and lower in contrast than normal

- If the sum of scene charge and smear charge saturates the pixels in the column, then the column looks like a bright defective column

- Smear usually affects several adjacent columns, so saturated columns become difficult to conceal well (high amount of defective data to conceal)

Smear Correction (2)

- Concealment approaches start with the use of dark rows or overclocked rows to estimate the smear signal that should be subtracted from each column
- For example, one may subtract a smear signal from each column and apply a gain adjustment after the subtraction
- Gain adjustment prevents bringing saturated columns down below the maximum code value, but adds gain variations to each column
- In general, high quality smear correction is very difficult to achieve

Outline

1. Exposure & Autofocus
   - Exposure
   - Autofocus
     - Active - Phase Detection
     - Passive - Contrast Detection
2. Colour Imaging Pipeline
   - Channel Matching
   - Dark Correction
   - Defect Concealment
   - Smear Correction
   - **Gain Nonuniformity Correction**
   - Optics Corrections
   - Stochastic Noise Reduction
   - Exposure and White Balance Correction
   - Demosaicing
   - Interpolation
     - Interpolating CFA Generated Data

Gain Nonuniformity Correction

- Gain non-uniformity is caused by lens effects like vignetting or sensor characteristics
- Correction is essentially a multiplication of each pixel with a gain map
- Early implementations, with very limited memory for storing gain corrections, used simple separable polynomials
- Later implementations stored small images, with a gain value for each colour channel for small tiles of the image
- These maps were often created to make the sensor response to a uniform illumination completely flat, which left taking lens effects and interactions uncompensated

- With increasing adoption of CMOS sensors and evolution to smaller pixels, gain corrections now usually include lens interactions
- For a camera with a fixed lens, these are relatively simple
- For cameras with interchangeable lenses, this creates new overhead to combine a sensor gain map with a lens interaction gain map

- If lens effects get too severe, gain correction is usually limited to minimise noise amplification
- Results in yet another system optimisation, trading off darkness versus noisiness in the corners

Outline

1 Exposure & Autofocus
- Exposure
- Autofocus
  - Active - Phase Detection
  - Passive - Contrast Detection

2 Colour Imaging Pipeline
- Channel Matching
- Dark Correction
- Defect Concealment
- Smear Correction
- Gain Nonuniformity Correction
- Optics Corrections
- Stochastic Noise Reduction
- Exposure and White Balance Correction
- Demosaicing
- Interpolation
  - Interpolating CFA Generated Data

Optics Corrections

- Common optics distortions: vignetting, geometric distortion, chromatic aberrations (longitudinal and lateral), and spatially varying reaction to an impulse light source

- Geometric distortion is corrected by warping the image to invert the change in magnification
- Extent of distortion is usually determined with calibration patterns like checkerboard images

- Lateral chromatic aberration is corrected similarly by applying the procedure to colour bands separately
- Longitudinal chromatic aberration (different color channels are focused at different distances from the lens) are treated by applying a sharpening filter to the affected colour bands
- As distortion correction may spatially resample the colour channels individually, it is often included in the processing chain after demosaicing

- Convolution with a spatially varying kernel is used to compensate for spatially varying reaction to an impulse light source

Outline

1 Exposure & Autofocus
- Exposure
- Autofocus
  - Active - Phase Detection
  - Passive - Contrast Detection

2 Colour Imaging Pipeline
- Channel Matching
- Dark Correction
- Defect Concealment
- Smear Correction
- Gain Nonuniformity Correction
- Optics Corrections
- Stochastic Noise Reduction
- Exposure and White Balance Correction
- Demosaicing
- Interpolation
  - Interpolating CFA Generated Data

Stochastic Noise Reduction (1)

- All noise reduction operations seek to preserve as much scene information as possible while smoothing noise
- To achieve this efficiently, it is important to use relatively simple models to discriminate between scene information and noise information

- In the stochastic noise reduction block, greyscale techniques for noise reduction are usually applied to each colour channel individually
- In addition, after demosaicing, inter-colourband correlation may be exploited to distinguish noise from structural scene information (“colour noise reduction”)

- The first technique applied is range based filtering:
  - This noise reduction is based on smoothing small intensity changes and retaining large ones
  - Problem: textures and edges with low contrast tend to get over-smoothed
  - The second artifact is the tendency to switch from smoothing to preservation when modulation gets larger:
  - Results in a very nonuniform appearance in textured fields or edges, with portions of the texture being smoothed and other portions being much sharper

Stochastic Noise Reduction (2)

- The second technique is based on the likelihood that impulses are noise:
  - Leads to the use of impulse filtering noise reduction
  - Usually using a standard center-weighted median filter
  - Characteristic artifact caused by impulse filtering is elimination of small details from the scene, especially specular reflections from eyes and small lights
  - When applying impulse filters to CFA data, the filtering is particularly vulnerable to creating colored highlights, if an impulse is filtered out of one or two colour channel(s)

Outline

1. Exposure & Autofocus
   - Exposure
   - Autofocus
     - Active - Phase Detection
     - Passive - Contrast Detection
2. Colour Imaging Pipeline
   - Channel Matching
   - Dark Correction
   - Defect Concealment
   - Smear Correction
   - Gain Nonuniformity Correction
   - Optics Corrections
   - Stochastic Noise Reduction
   - Exposure and White Balance Correction
   - Demosaicing
   - Interpolation
     - Interpolating CFA Generated Data

HVS:

- Has the ability to map "white" colours to the sensation of white
- Even though an object has different radiance when it is illuminated with different light sources
- In other words, a sheet of white paper under fluorescent lighting or under incandescent lighting or even under natural daylight appears to be white, although the actual irradiated energy produces different colors for different illuminations
- This phenomenon is called *color constancy*

DSC and SLR:

- Need to be taught how to map white under the capture illuminant to white under the viewing illuminant (and other colours accordingly)

- White balance adjustment is accomplished by multiplying pixels in each colour channel by a different gain factor that compensates for a non-neutral camera response and illuminant imbalance
- Application of the gain factors to the CFA data before demosaicing may be preferred:
  - Some demosaicing algorithms may presume equal responses for the different colour channels

- Camera's response to typical illuminants, such as daylight, incandescent, and fluorescent, is easily stored in the camera

- In some circumstances, the capture illuminant is known (or can be determined)
- For example this is the case for flash usage or for user controlled illuminant selection on the camera

- Another option to determine illuminant is to consider several possible illuminant classes and estimate the probability of each illuminant being the actual scene illuminant based on the colour characteristics

Automated White Balance (AWB):

- In most cases, it is desirable to perform white balance without knowledge about the capture illuminant
- Appropriate gain factors need to be determined to correct for illumination imbalance
- Current cameras approach this estimation problem with different algorithms having different responses to scene content and illuminants
- Camera manufacturers usually have somewhat different preferences:
  - E.g. biasing white balance to render images warmer or cooler
  - Also different approaches to estimating the scene illuminant

Exposure and White Balance Correction (3)

- Best way to do white balance is to take a picture of a neutral object (white or gray) and deduce the weight of each channel:
  - If the object is recorded as R_w, G_w, B_w
  - Use weights $1/R_w, 1/G_w, 1/B_w$ for the three colour channels

- One means of performing auto white balance is to assume that a white patch must induce maximal camera responses in the three channels
- The underlying theory is that highlights are specular reflections that are the colour of the illuminant
- Thus, the white-balanced image has signals given by $R/R_{max}, G/G_{max}, B/B_{max}$
- However, the maximum in the three channels is very often a poor estimate of the illuminant and it does not work for scenes that have no truly specular highlights

- Most automatic white balance and exposure algorithms are based on some extension of the gray world model:
  - Assume all colours in an image will average out to gray, $R = G = B$
  - Using this approach, the channels are scaled based on the deviation of the image average from gray
  - In this scheme, the white-balanced image has signals given by:

$$k_r * R, G, k_b * B$$

  where $k_r = G_{mean}/R_{mean}$ and $k_b = G_{mean}/B_{mean}$

- However, the actual gray world model assumes that images of many different scenes will average out to 18% gray (a midtone gray)
- Unfortunately, this says very little about a specific image, but the algorithm must work well for individual images
- Therefore, most extensions of the gray world model try to disregard large areas of single colours (not taking them into account for the average gray calculation):
  - Avoid having the balance driven one way or another by red buildings, blue skies, or green foliage

Outline

1. Exposure & Autofocus
   - Exposure
   - Autofocus
     - Active - Phase Detection
     - Passive - Contrast Detection
2. Colour Imaging Pipeline
   - Channel Matching
   - Dark Correction
   - Defect Concealment
   - Smear Correction
   - Gain Nonuniformity Correction
   - Optics Corrections
   - Stochastic Noise Reduction
   - Exposure and White Balance Correction
   - Demosaicing
   - Interpolation
     - Interpolating CFA Generated Data

- Process of generating three equally populated colour bands with full resolution from an image captured using a CFA technique
- For this purpose, artificial data needs to be generated since all three colour bands are available in sub-sampled form:
  - I.e. the green channel has 50% and the red and blue channels have 25% of pixels populated, respectively
- The technique used to generate these missing data is called *interpolation*:
  - Apart from demosiacing, interpolation is used in:
    - Image resizing/scaling
    - Defect concealment / correction (image impairment)
    - Super-resolution
    - Many other techniques...
  - Due to its importance, we first shed some light on basic principles of interpolation

Outline

1. Exposure & Autofocus
   - Exposure
   - Autofocus
     - Active - Phase Detection
     - Passive - Contrast Detection
2. Colour Imaging Pipeline
   - Channel Matching
   - Dark Correction
   - Defect Concealment
   - Smear Correction
   - Gain Nonuniformity Correction
   - Optics Corrections
   - Stochastic Noise Reduction
   - Exposure and White Balance Correction
   - Demosaicing
   - Interpolation
     - Interpolating CFA Generated Data

Demosaicing - Interpolation Basics

- Classical interpolation is the process to compute an interpolated value $g(x)$ at some (perhaps non-integer) coordinate x as a linear combination of the samples g_k evaluated at integer coordinates k
- The weights being given by the values of the function $f(x-k)$:

$$g(x) = \sum_{k \in Z} g_k f(x-k) \ .$$

- $f(x)$ must vanish for all integer arguments except at the origin, where it must be 1 (i.e. "interpolation property")
- Summation is performed over all integer coordinates
- However, in practice the number of known (or used) samples is always finite

- A large variety of different "interpolation kernels" $f(x)$ is used
- Having different properties with respect to resulting quality of the interpolated data, execution speed of the computation, memory requirement etc.

Nearest neighbour kernel is the simplest of all:

- $f_{NN}(x) = 1$ for $-0.5 \leq x < 0.5$
- $f_{NN}(x) = 0$ if $x < -0.5$ and $x \geq 0.5$

- For any coordinate x where it is desired to compute the value of the interpolated function g, there is only one sample g_k that contributes
- Main interest of this approach is its simplicity, the price to pay is a very low quality

Linear interpolation still offers very low complexity but improves quality as compared to nearest neighbour interpolation considerably:

- $f_{LIN}(x) = 1 - |x|$ for $|x| < 1$
- 0 otherwise ($|x| \geq 1$)

- How does this correspond to our usual notion of taking the sum and divide by two ?
- For example, consider two pixel values (6 and 10) next to each other and we want to compute the interpolated value right in the middle of them
- Following our general formula, we result in:

$$g(0) = 10 f_{LIN}(-0.5) + 6 f_{LIN}(0.5) = 5 + 3 = 8 \; .$$

- This is exactly the result we expect.

- In two dimensions, also called bilinear interpolation, its separable implementation requires four samples
- Here, first columns are interpolated, followed by an interpolation of the lines

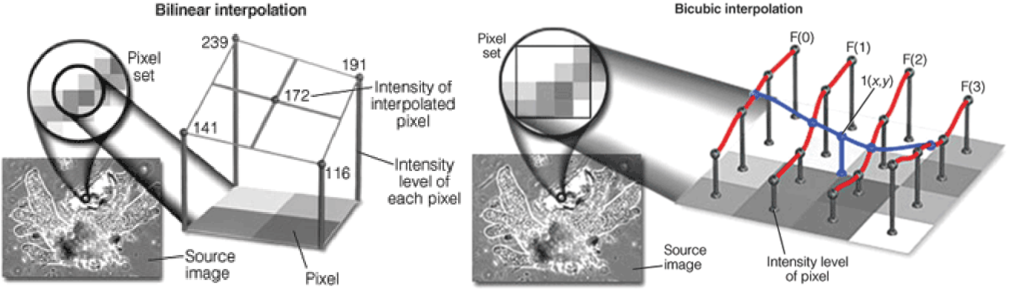


Figure: Bilinear and Bicubic interpolation

Cubic interpolation:

- Produces less blurring of edges and other distortion artifacts than bilinear interpolation
- But is more computationally demanding
- Polynomials of third degree are used as kernel functions such that more sample points can be considered

- Bicubic interpolation involves fitting a series of cubic polynomials to the pixels contained in a 4×4 array of pixels surrounding the calculated address
- First, four cubic polynomials are fitted to the control points in the y-direction (the choice of starting direction is arbitrary)
- Next, the fractional part of the calculated pixel's address in the y-direction is used to fit another cubic polynomial in the x-direction, based on the interpolated pixel values that lie on the curves
- Substituting the fractional part of the calculated pixel's address in the x-direction into the resulting cubic polynomial then yields the interpolated pixel's brightness value

- Bicubic interpolation has found use in many commercial software packages such as Adobe Photoshop and others

Demosaicing - Cubic Interpolation - Deriving a Kernel Function (1)

The choice of polynomial used in the (bi)cubic interpolation algorithm can have a significant impact on the accuracy and visual quality of the interpolated image.

In the following, we demonstrate how to derive a cubic interpolation kernel function $f_{CUB}(x) = f(x)$:

- If the values of a function and its derivative are known at $x = 0$ and $x = 1$
- Then the function can be interpolated on the interval $[0, 1]$ using a third degree polynomial:
- $f(x) = ax^3 + bx^2 + cx + d$, $f'(x) = 3ax^2 + 2bx + c$
- The values of the polynomial and its derivative at $x = 0$ and $x = 1$ are given as:
- $f(0) = d$, $f(1) = a + b + c + d$, $f'(0) = c$, and $f'(1) = 3a + 2b + c$
- The four equations can be rearranged so that they deliver the required polynomials' coefficients:
- $a = 2f(0) - 2f(1) + f'(0) + f'(1)$, $b = -3f(0) + 3f(1) - 2f'(0) - f'(1)$, $c = f'(0)$, and $d = f(0)$

Problem:

- In most cases (particularly in image processing), we do not know the derivative of the underlying (image intensity) function
- We simply want to interpolate between a list of pixels

- Instead of setting the derivative to 0 at each point (which does not lead to smooth curves)
- We use the slope of a line between the previous and the next point as the derivative at a point (the resulting kernel is called "a Catmull-Rom spline")

- Suppose we have the samples g_0, g_1, g_2, and g_3
- At the positions $x = -1$, $x = 0$, $x = 1$, and $x = 2$
- Then we can assign the values of $f(0)$, $f(1)$, $f'(0)$ and $f'(1)$ using the formulas below to interpolate between:
- g_1 and g_2: $f(0) = g_1$, $f(1) = g_2$, $f'(0) = \frac{g_2 - g_0}{2}$, and $f'(1) = \frac{g_3 - g_1}{2}$
- Setting these values into the above formula for the polynomial coefficients we result in:
- $a = -1/2g_0 + 3/2g_1 - 3/2g_2 + 1/2g_3$, $b = g_0 - 5/2g_1 + 2g_2 - 1/2g_3$, $c = -1/2g_0 + 1/2g_2$, and $d = g_1$
- Resulting in the corresponding polynom: $f(g_0, g_1, g_2, g_3, x)$

- For bicubic interpolation, suppose we have the 16 samples (pixels):
- g_{ij} with i and j going from 0 to 3 and with g_{ij} located at $(i-1, j-1)$
- Then we can interpolate the area $[0,1]^2$:
- By first interpolating the four columns and
- Then interpolating the results in the horizontal direction
- The formula for the polynom becomes:

$$\begin{aligned} f(x,y) &= f(f(g_{0,0}, g_{0,1}, g_{0,2}, g_{0,3}, y), f(g_{1,0}, g_{1,1}, g_{1,2}, g_{1,3}, y), && (16) \\ &\quad f(g_{2,0}, g_{2,1}, g_{2,2}, g_{2,3}, y), f(g_{3,0}, g_{3,1}, g_{3,2}, g_{3,3}, y), x) \,. && (17) \end{aligned}$$

- Alternatively, the formula can be derived if the following function values are known:
- $f(x,y)$, $f_x(x,y)$, $f_y(x,y)$, and $f_{xy}(x,y)$ at the four corners $(0,0)$, $(1,0)$, $(0,1)$, and $(1,1)$
- The unknown coefficients a_{ij} of the corresponding 2-D polynomial surface $f(x,y) = \sum_{i=0}^{3} \sum_{j=0}^{3} a_{ij} x^i y^j$ can be computed by solving a system of 16 linear equations, similar to the procedure above for the one dimensional case

- Compute an interpolation polynomial (the green curve) for the four points on the red curve
- The four points $g_0 = 2$, $g_1 = 4$, $g_2 = 2$, and $g_3 = 3$, at the positions $x = 1$, $x = 2$, $x = 3$, and $x = 4$ are given (note that the x-positions are different compared to those used in the derivation)

We compute the resulting polynomials' coefficients as:

- $a = -1/2 * 2 + 3/2 * 4 - 3/2 * 2 + 1/2 * 3$
- $b = 2 - 5/2 * 4 + 2 * 2 - 1/2 * 3$
- $c = -1/2 * 2 + 1/2 * 2$
- $d = 4$

- Resulting in the polynom:
- $f(x) = 7/2(x-2)^3 - 11/2(x-2)^2 + 4$
- Note: x-2 replaces x due to the shift from $[0, 1]$

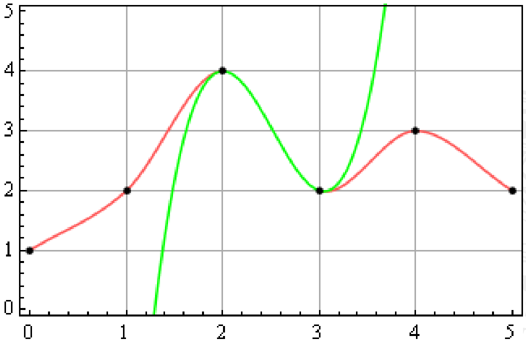


Figure: Example for cubic interpolation

Bicubic Spline Interpolation:

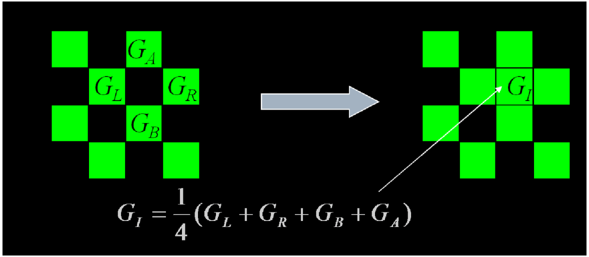
- Requires the solution of the linear system described above for each grid cell
- A fixed kernel with similar properties is often used instead (as derived by Keys):
- $f_{KEYS}(x) = (a+2)|x|^3 - (a+3)|x|^2 + 1$ for $0 \leq |x| < 1$
- $f_{KEYS}(x) = a|x|^3 - 5a|x|^2 + 8a|x| - 4a$ for $1 \leq |x| < 2$
- $f_{KEYS}(x) = 0$ for $|x| \geq 2$
- Often, a fixed choice is $a = 0.5$

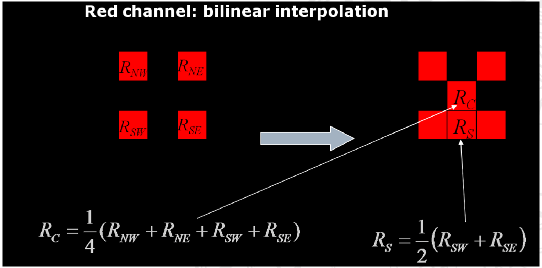
Many more interpolation kernels do exist:

- Lanczos kernel
- Sinc kernel
- Various types of spline interpolation methods

- An important issue for these algorithms is computational cost and ease of hardware implementation
- It has to be noted that many of the techniques described are covered by patents of the respective camera producers
- Often, the actual technique used in a camera is not publicly known

- The first and most obvious approach is to apply interpolation techniques to each colour plane independently
- Nearest neighbour interpolation makes an arbitrary choice which pixel is selected for identical distance, bilinear interpolation averages between the neighbours:





- The examples below illustrate that significant colour artifacts arise with this strategy
- The effect displayed is called “Colour Moire effect (or colour fringes or zipper effect)” and is caused by misinterpreting luminance detail as colour information
- Note that also when applying more advanced interpolation (like bicubic techniques), those effects cannot be reduced significantly

Figure: Examples for colour plane interpolation: nearest neighbour vs. bilinear

- Caused by the poor interpolation results of individual colourplane interpolation:
- Sharp luminance transitions cause a sharp transition in the colour planes at different spatial locations
- I.e. the colour planes do not react in a synchronized manner to sharp edges
- An example of this effect and the situation causing the effect is shown below:

Blow-up of electronic camera image. Notice spurious colors in the regions of fine detail in the plants.

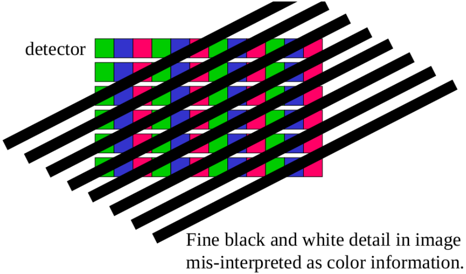


Figure: Colour Moire artefact

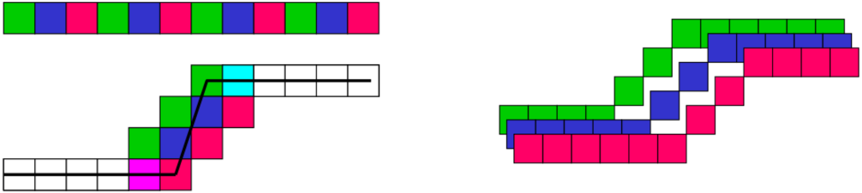
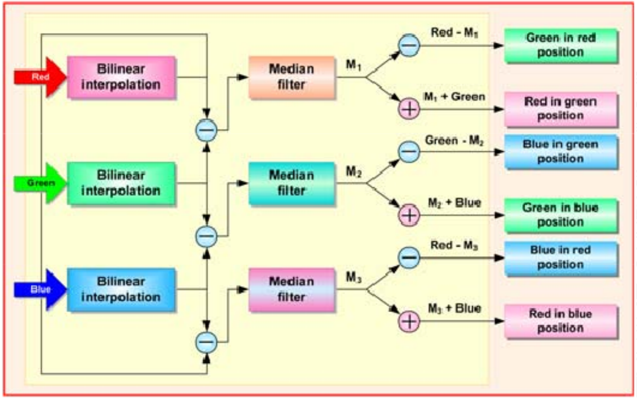


Figure: Principle of color sampling errors

- The figure above illustrates what exactly happens at a sharp luminance transition
- As a consequence, it is imperative to incorporate the inter-colourband correlations into the demosaicing process
- A significant number of corresponding approaches have been suggested throughout the last 2 decades
- The tendency is to increase complexity resulting in steadily increasing quality in this field

The first approach employs a *median filter* to colourplane differences. The idea is clear:

- Since the colourplanes are out of synchronisation, a difference signal contains isolated maximas in areas where colour fringe occurs (see figure left for the R-G signal)
- Therefore, as illustrated, a median filter is applied to colour difference signals, the results of which are used with original measurements to compute all the RGB values in each pixel
- This is possible as we have one value and two differences for each pixel



Example of a filter's difference signal (R-G signal) and a comparison of the colourplane independent bilinear interpolation (middle) and the median filtering approach (right). The latter one (median filtering approach) shows clearly reduced colour artefacts.

Figure: Median filtering, left: R-G signal, centre: bilinear interpolation, right: median filtering result

- Median filtering approach is a first approach to take into account the strong spectral correlation between colour components at each pixel
- Two main hypotheses are proposed in the literature in this context:
  - The first one assumes a *color ratio constancy*
  - The second one is based on *color difference constancy* (where median filtering obviously relies on the latter)

Interpolation based on colour hue constancy:

- Follows the first idea, where *hue* is understood as the ratio between chrominance and luminance
- I.e. R/G or B/G when the G plane is identified with luminance as it is often done
- Exibits problems in case the denominator G takes low values
- This happens for instance when saturated red and/or blue components lead to comparatively low values of green, making the ratios R/G and B/G very sensitive to red and/or blue small variations


The image above is highly saturated in red (left) with its corresponding G plane (right). The figure below shows the component ratio R/G (left) and difference R-G (right). Note that these two images carry out less high-frequency information than the green component plane.


- A Sobel filter is then applied to these two images, so as to highlight the high-frequency information location
- In the right-hand parrot plumage area where red is saturated, the component ratio plane (left) contains more high-frequency information than the component difference plane (right)
- This makes it more artifact-prone when demosaiced by interpolation
- Moreover, high colour ratio values may yield to estimated component levels beyond the data bounds
- $\rightarrow$ undesirable for the demosaicing result quality

Constant hue transition interpolation method:

- First interpolates the G plane by some desired method (by-linear or edge-directed, see below).
- Using the assumption that hue is smoothly changing across an objects surface, the hue value is interpolated
- The interploation for the chrominance values are derived from the interpolated hue values
- To be more specific, the interpolated R hue (R/G ratio) and B hue (B/G ratio) are multiplied by the G value to determine the missing R and B values at a given pixel position
- For example, refering to the Bayer pattern in the figure on the next slide, the following formulas are used:

$$R_{44} = G_{44} \frac{\frac{R_{33}}{G_{33}} + \frac{R_{35}}{G_{35}} + \frac{R_{53}}{G_{53}} + \frac{R_{55}}{G_{55}}}{4}$$

$$B_{33} = G_{33} \frac{\frac{B_{22}}{G_{22}} + \frac{B_{24}}{G_{24}} + \frac{B_{42}}{G_{42}} + \frac{B_{44}}{G_{44}}}{4}$$

- Note that the G values involved in these formulas are the result of the first interpolation

The figure below illustrates how this concept can be used employing colourband differences instead of hue:

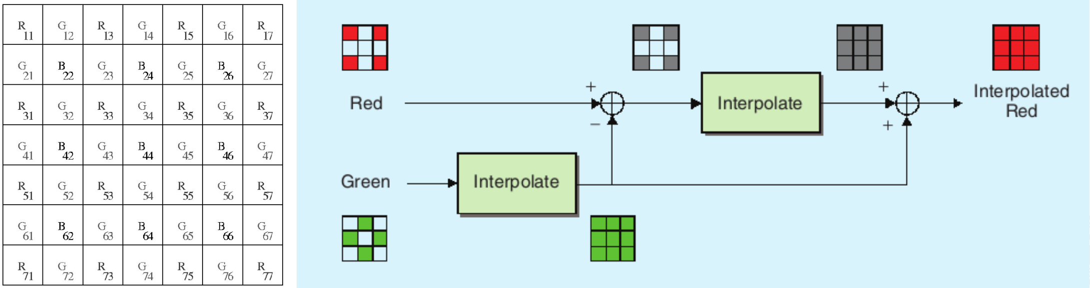
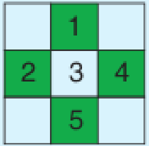


Figure: Bayer CFA pattern and constant-difference-based interpolation

- Note: all of the afore mentioned demosaicing algorithms are **nonadaptive**.
- Nonadaptive demosaicing algorithms typically provide satisfactory results in smooth image regions, while they usually fail in textured regions and edges

Edge-directed interpolation:

- Adaptive approach, where the area around each pixel is analysed to determine if a preferred interpolation direction exists
- In practice, the interpolation direction is chosen to avoid interpolation across edges
- Instead interpolation is performed along any edges in the image
- Figure below shows an example of applying this idea to a single colour band
- It can also be applied to any grayscale image and is therefore also a generic adaptive interpolation approach
- In practice, the gradients themselves and their difference should exceed some threshold
- The idea can also be combined also with bicubic or any other more advanced technique



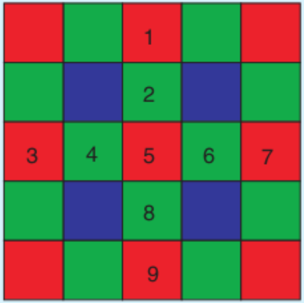
1. Calculate horizontal gradient $\Delta H = |G2 - G4|$
2. Calculate vertical gradient $\Delta V = |G1 - G5|$
3. If $\Delta H > \Delta V$,
   $G3 = (G1 + G5)/2$
   Else if $\Delta H < \Delta V$,
   $G3 = (G2 + G4)/2$
   Else
   $G3 = (G1 + G5 + G2 + G4)/4$

- This idea can be extended to exploit inter-colourband correlation as well (see figure on the next slide)
- Here, the R and B channels in a larger neighbourhood are used instead of the G channel to determine gradients, second-order derivatives are used
- Once the luminance is determined, chrominance values are are interpolated from the differences bewteen the colour (R and B) and luminance (G) channels (again ratios can be used as well)
- For example (notation of figure below is used):

$$R8 = \frac{(R5 - G5) + (R9 - G9)}{2} + G8 \text{ and } R4 = \frac{(R3 - G3) + (R5 - G5)}{2} + G4\,.$$

- For the red value in a blue pixel the four differences NW, NE, SW, and SE are added, divided by four, and the corresponding G interpolation value is added

Demosaicing - Adaptive Demosaicing - Edge-Directed Interpolation (3)



1. Calculate horizontal gradient $\Delta H = |(R3 + R7)/2 - R5|$
2. Calculate vertical gradient $\Delta V = |(R1 + R9)/2 - R5|$
3. If $\Delta H > \Delta V$,

   $G5 = (G2 + G8)/2$

   Else if $\Delta H < \Delta V$,

   $G5 = (G4 + G6)/2$

   Else

   $G5 = (G2 + G8 + G4 + G6)/4$

Figure: Edge-directed interpolation involving all colour planes.

Adaptive colour plane interpolation:

- Improves the previous approach by also using colour plane information to interpolate the green band, i.e. (notation of previous figure is used):

$$G5 = \frac{G2 + G8}{2} + \frac{2 * R5 - R1 - R9}{2} \text{ for } \delta V < \delta H \ ,$$

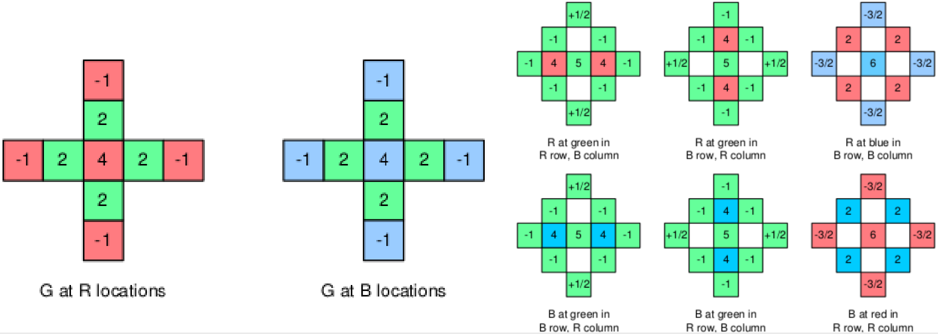
$$G5 = \frac{G4 + G6}{2} + \frac{2 * R5 - R3 - R7}{2} \text{ for } \delta V > \delta H \text{ , and}$$

$$G5 = \frac{G2 + G4 + G6 + G8}{4} + \frac{4 * R5 - R1 - R3 - R7 - R9}{4} \text{ for } \delta V = \delta H \ .$$

- Here, in fact second order colour gradients are used in the interpolation
- In the original scheme, also δV and δH are more complicated
- The colour channels are interpolated using a similar technique
- A further refinement is to use more directions for computing gradient information

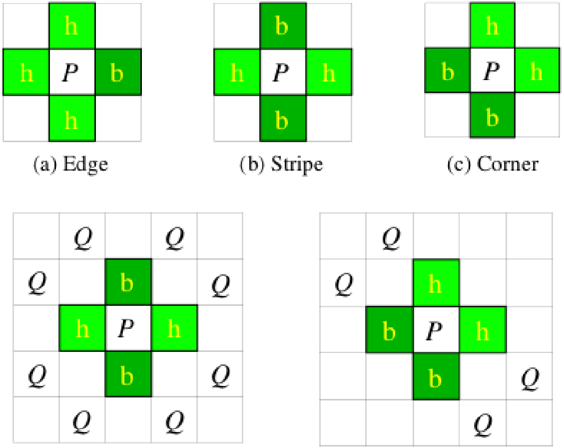
High Quality Linear Interpolation:

- Very similar to previous one but uses different weights in its interploation scheme
- 8 different cases are distinguished:
  - 2 to determine the red and green values on a blue pixel
  - 2 to determine the blue and green values on a red pixel
  - 4 to determine the red and blue values on a green pixel (2 for a green pixel in a "red row" and 2 in a "blue row")
- Figure below shows the corresponding 8 interpolation schemes (which need to be normalised before application); e.g. used in MATLAB



Pattern Recognition Interpolation:

- This family of methods aims at identifying a template-based feature in each pixel neighborhood in order to interpolate according to the locally encountered feature
- The first step in his procedure is to find the average of the four neighboring green pixels, and classify the neighbours in comparison to this average as either:
  - high (h)
  - or low (b)
- See figure to the right for the patterns used to determine the central pixel interpolation



(a) Edge (b) Stripe (c) Corner

(d) Stripe neighborhood (e) Corner neighborhood

- These values are sorted and denoted as $G_1, \ldots, G_4$, $M = \frac{G_2+G_3}{2}$
- The green pixel $\hat{G}$ is then defined as an edge if three neighbor pixels share the same classification
- If not, then the pixel can either be a part of a corner or a stripe
- If two adjacent neighbour pixels have the same classification, then the pixel is a corner
- If two opposite pixels have the same classification, then the pixel is a stripe
- If an edge is detected: $\hat{G} = M$
- For a stripe: $\hat{G} = CLIP(M - (S - M))$ where S is the average green level over the eight neighboring pixels labeled as Q in the figure
- For a corner: $\hat{G} = CLIP(M - (S' - M))$ where S' is the average green level over the eight neighboring pixels labeled as Q in the figure
- $CLIP$ limits the interpolated value to $[G_3, G_2]$
- The other colour planes can be interpolated using any of the techniques described before

Homogeneity-directed interpolation:

- The RGB data is first interpolated horizontally and vertically, i.e., there are two candidates for each missing color sample
- Both the horizontally and vertically interpolated images are transformed to the CIELAB space
- In the CIELAB space, either the horizontally or the vertically interpolated pixel values are chosen based on the local homogeneity
- The local homogeneity is measured by the total number of similar luminance and chrominance values of the pixels that are within a neighborhood of the pixel in question

Vector-based interpolation:

- Each pixel is considered as a vector in the three dimensional (R,G,B) space
- Interpolation is designed to minimise the angle or the distance among neighbouring vectors
- After an inital interpolation of missing samples, each pixel is transformed to spherical coordinates (ρ, Φ, ϕ):

$$R = \rho \cos(\Phi) \sin(\phi) \ , \ R = \rho \cos(\Phi) \cos(\phi) \ , \ B = \rho \sin(\Phi) \ .$$

- In the (ρ, Φ, ϕ) space, a filtering operation like median filtering is applied to the angles only
- This forces the chrominance components to be similar
- Because ρ is closely related to the luminance component, keeping it unchanged preserves the luminance discontinuities among neighboring pixels
- After the filtering process, the image is transformed back to the (R, G, B) space