* Machine vision :- It is the technology & method used to provide imaging based automatic inspection & analysis for such application as automatic inspection.

* Teach pendent robot :- It is need to control an industrial robot remotely robot control step by step by remote they are ocupied with with switches & dials to control robot

Machine Vision Algorithms

* Fundamental data structure :- That are involved in machine vision applications. The data structures for images, regions & subpixel-precise contours.

1. **Images :-**

An image is the basic data structure in machine vision, since this is the data that an image acquisition device typically delivers to the computer's memory.

A pixel can be regarded as a sample of the energy that falls on the sensor element during the exposure, integrated over the spectral distribution of the light & the spectral response of the sensor.

Depending upon camera type, the spectral response of the sensor typically the entire visible spectrum & optionally a part of the near IR spectrum.

In case, camera will return one sample of the energy per pixel. i.e. single channel grey value image.

RGB camera will return three sample per pixel i.e. Three channel image.

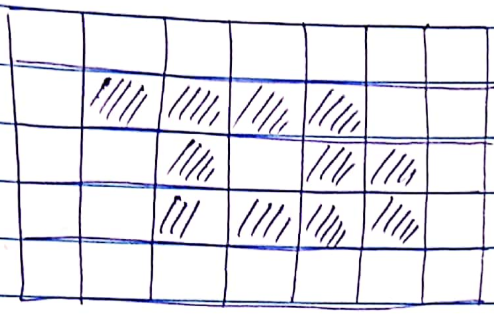
These are the two basic type of sensors that are encountered in machine vision appl^n.

Image channel regarded 2D array of numbers.

This is also data structure that is used to represent images in a programming language.

Hence, the grey value at the pixel $(r,c)^T$ can be interpreted as an entry in a matrix : $g = f_{r,c}$.

eg:-



Run	Row	column Start	column End
1	1	1	4
2	2	2	2
3	2	4	5
4	3	2	5

fig:- Run-length representation of a region.

- Either horizontally or vertically, there are extended runs in which adjacent pixels belong to the region.
- This is typically the case for most region.
  Use this property & store only the necessary data for each run.
- since images are typically stored line by line in memory, it is better use horizontal run.
- Therefore, the minimum amount of data for each run is the row coordinate of the run & start & end column of the run.
- This method of storing a region is called a run-length representation or run length encoding.
- consequently, the region can also be regarded as the union of all of its runs.

$$R = \bigcup_{i=1}^{n} r_i$$

Here r_i → a single run → regarded as region

2. Regions :-

The minimum a representation for an arbitary subset of the pixels in an image, for morphological op^n.
Region can also extend beyond the image borders to avoid artifacts. ∴ A region as an arbitrary subset of the discrete plane : $R \subset Z^2$

R is intentionally identical to the R that is used in the rectangle of image.

It is extremely useful to restrict processing to a certain part of the image that is specified as region of interest (ROI)

an image as function from the ROI to a set of numbers i.e. $f: R \rightarrow R^n$
it is also called as domain of image function f.

An abstract point of view, not immediately clear, how best to represent region.

Mathematically, region are set

An equivalent def^n is to use the characteristic f^n of the region :

$$X_R(r,c) = \begin{cases} 1 & (r,c) \in R \\ 0 & (r,c) \notin R \end{cases}$$

Use of Binary images to represent regions.
Where 1 = Included in region.
0 = Not include in region

3. Subpixel - Precise contours :-

- data structure considered are pixel-precise.
- it is important to extract subpixel - precise data from an image because the application requires an accuracy that is higher than the pixel resolution of the image.
- The subpixel data can.
  eg:. ① subpixel thresholding.
  ② subpixel edge extraction.
- The result of these operations can be described with subpixel - precise contours.

eg:- Contours can basically be represented as a polygon i.e. ordered set of control point $(r_i, c_i)^T$

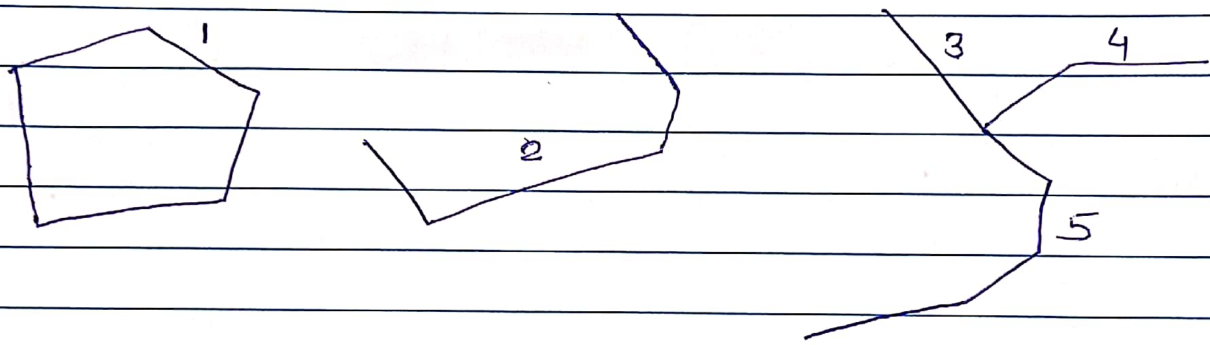


Fig:. Different subpixel - precise contours
Contour 1 is a closed contour
Contour 2-5 are open contour.
Contour 3-5 meet at a junction point.

Image Enhancement :-

The illumination, lenses, cameras, & image acquisition devices to obtain a good image quality.

Try to select best possible h/w setup, sometimes the image quality is not sufficient.

∴ We will take a look at several common techniques for image enhancement.

1. Gray Value Transformation :-

Controlling the illumination, it is necessary to modify the gray value image. This may be weak contrast.

With controlled illumination, this problem usually only occurs locally. Therefore need to increase the contrast locally.

Another possible reason for adjusting the gray value may be that the contrast or brightness of the image has changed from the setting that were in effect when set up application.

A gray value transformation can be regarded as a point operation.

Transformed gray value $t_{r,c}$ depends only on the gray value $g_{r,c}$ in the input image at the same position : $t_{r,c} = f(g_{r,c})$

Where $f(g)$ is a f^n that define the gray value T^x to apply domain & range of $f(g)$ typically G_b i.e. they are discrete.

∴ ↑ transformation speed, gray value T^x can be implement as a lookup table (LUT) by storing the o/p gray value for each possible i/p gray.

If we denote LUT as f_g,

$$t_{r,c} = f_g[g_{r,c}]$$

Where the [ ] operator denotes table look-up.

→ Contrast enhancement :-

GVT is a linear gray value scaling :

$$f(g) = ag + b$$

Hence clip & round the o/p gray value as follow

$$f(g) = \min(\max(\lfloor ag + b + 0.5 \rfloor, 0), 2^b - 1)$$

for $|a| > 1$ ..... The contrast is ↑

While for $|a| < 1$ ..... The contrast is ↓

if $a < 0$ .... gray value are inverted

For $b > 0$ .... The brightness is ↑

While for $b < 0$ ...... The brightness is ↓

→ Contrast Normalization :-

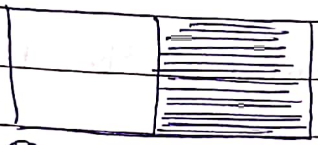
Select the parameter such that the maximum range of the gray value space G_b is used. This can be done as follows :

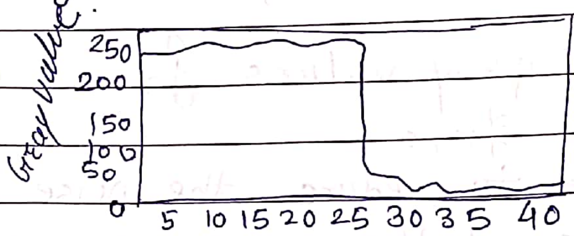
Let g_{min} & g_{max} be the minimum & maximum gray value in the ROI under consideration. (Region of interest)

Then, the maximum range of gray values will be used if $a = (2^b - 1) / (g_{max} - g_{min})$ & $b = -a g_{min}$

This transformation can be thought of as a normalization of the contrast

Image smoothing :-

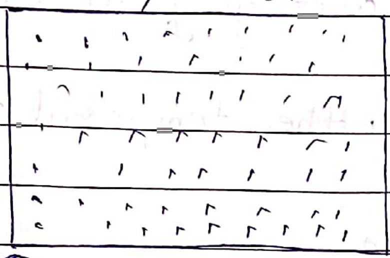


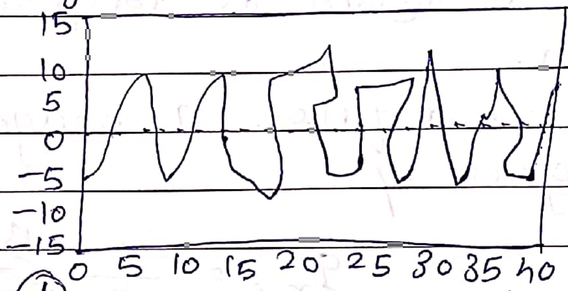


ⓐ An image of an edge. from real appln

ⓑ Position along profile (Pixel)

The noise is clearly visible in the bright patch.





ⓒ The noise in ⓐ Scaled by a factor of 5

ⓓ Horizontal grey value profile of the noise.

✓ Every image contains some degree of noise.

✓ The true grey value $g_{r,c}$ is disturbed by noise $n_{r,c}$ to get the observed grey value.

$$\hat{g}_{r,c} = g_{r,c} + n_{r,c}$$

✓ The noise $n_{r,c}$ as a random variable with mean 0 & variance σ^2 for every pixel.

✓ The noise has been calculated is explained below.

✓ There is slightly more noise in the dark patch of the image.

1. Temporal averaging :-

- Noisy gray values $\hat{g}_{r,c}$, estimate the true gray value $g_{r,c}$.
- Method to reduce the noise is to acquire multiple images of the same scene & to simply average these images. Since images are taken at different time, This method as temporal averaging or temporal mean.
- If we acquire n images, the temporal average is given by:

$$g_{r,c} = \frac{1}{n} \sum_{i=1}^{n} \hat{g}_{r,c;i}$$

Where $\hat{g}_{r,c;i}$ denotes the noisy gray value at position $(r,c)^T$ in image i.

- The variance of the noise is reduced by a factor of n by this estimation.

$$\sigma_m^2 = \sigma^2 / n$$

- consequently, the standard deviation of the noise is reduced by a factor of $\sqrt{n}$.

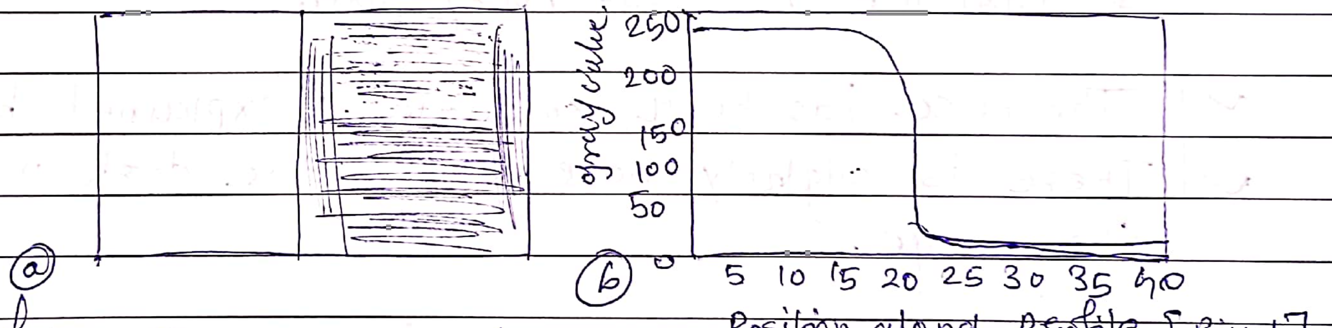


fig: (a) An image of an edge obtained by averaging 20 images of the edge.
(b) Horizontal gray value profile through the center of the image.

★ Drawback :-
- less speed
- Acquire multiple images to reduce the noise.

Spatial data — geographic data

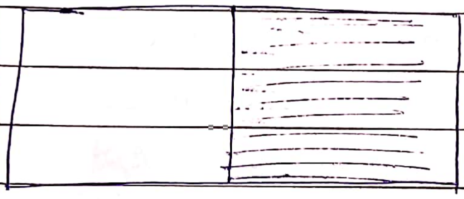
2. <u>Mean filter</u> :-

- ✓ means for reducing the noise are required.
- ✓ Ideally, use only one image to estimate the true grey value.
- ✓ The spatial avg or spatial mean can be computed over a window of.

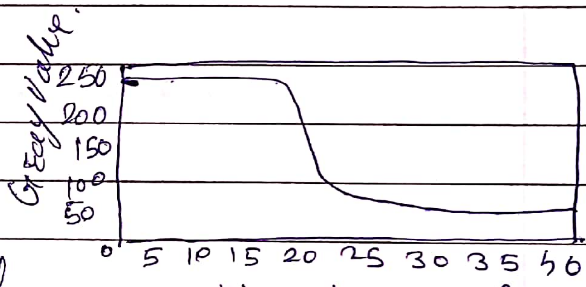
(2n+1) × (2m+1) Pixel as follows :-

$$g_{r,c} = \frac{1}{(2n+1)(2m+1)} \sum_{i=-n}^{n} \sum_{j=-m}^{m} \hat{g}_{r-i,c-j}$$

The spatial averaging operation is also called a mean filter.

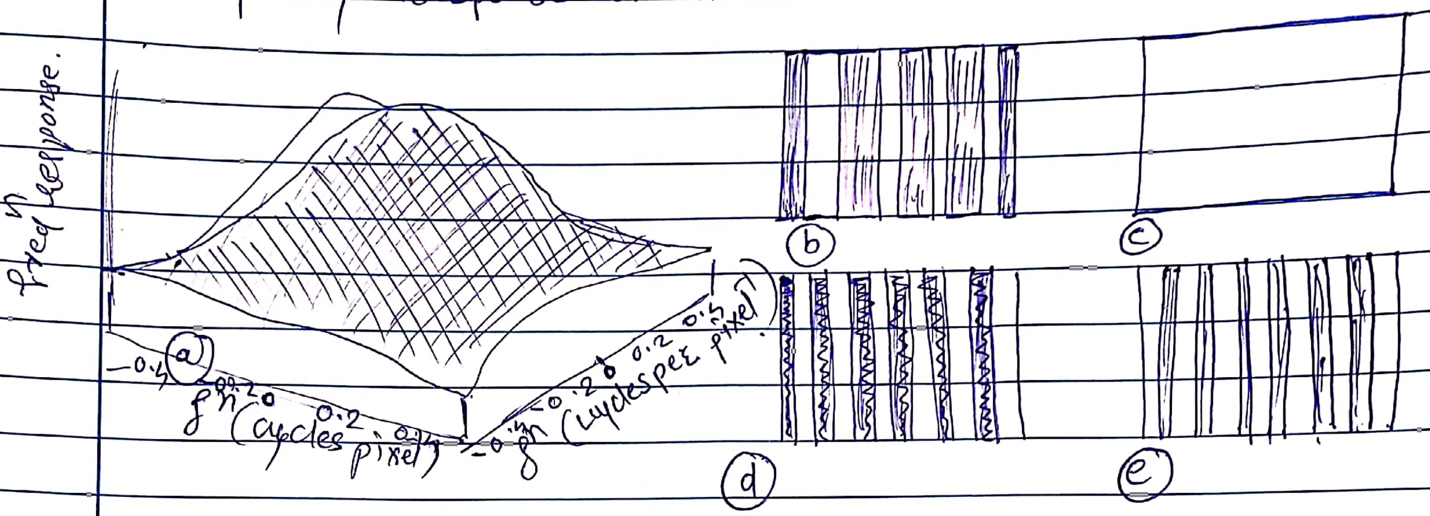


(a) An image of an edge obtained by smoothing the image $\bar{c}$ a 5×5 mean filter.



(b) Horizontal grey value profile through the center of the image.

3) Frequency response of the mean filter.



- Although the mean filter produce good result, it is not the optimum smoothing filter.
- Noise primarily manifests itself as high-frequency fluctuation of the gray value in the image.
- Ideally, a smoothing filter to remove these high frequency fluctuation.
- Mean filter perform this task. Mean filter respond to certain f^n in the image.
- Fig (a) The frequency response of a 3 X 3 mean filter
- In this plot row & column coordinate represen the f^n as cycles per pixel.
- If both coordinates are 0, this correspond to f^n of 0 cycles per pixel., which represent the average gray value in the image.
- row & column coordinate of ± 0.5 represent the highes possible f^n in the image (one cycle per two pixel)

(a) 3X3 mean filter removes certain f^n completely. These are point for which the response has a value of 0. They occur for relatively high frequencies. see that highest f^n are not removed completely

(b) An image with one pixel wide line spaced three pixel apart

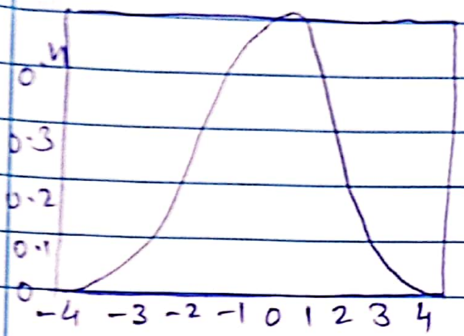
(c) This f^n is completely remove by the 3X3 mean filter The o/p image has a constant grey value.

(d) If we change the spacing of line to two pixel fig (d) That this higher f^n is not removed completely This is an undesirable behavior since it means that noise is not removed completely by mean filter.

✓ The Polarity of line has been reversed by mean filter. Which is also undesirable. This is cause by the negative part of the f^n response.

fig (a) The f^n response of mean filter is not rotationally symmetric i.e. it is anisotropic
This mean diagonal structures are smoothed differently than horizontal or vertical structures.

Gaussian filter :-



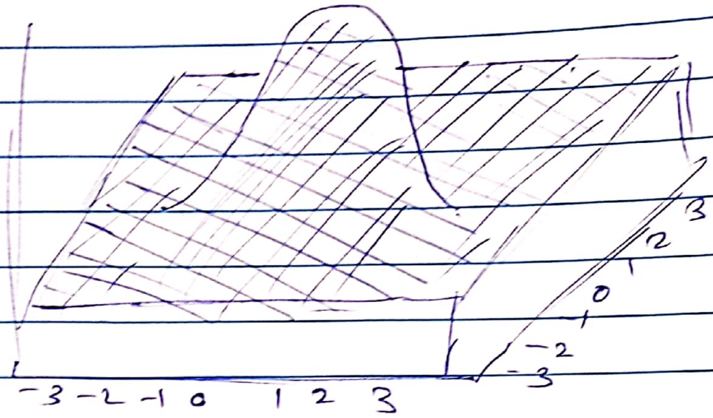


fig: (a) 1D Gaussian filter with $\sigma=1$ (b) 2D Gaussian filter with $\sigma=1$

In one dimension, the Gaussian filter is given by

$$g_\sigma(x) = \frac{1}{\sqrt{2\pi}\,\sigma} e^{-x^2/2\sigma^2}$$

This is the function that also defines the probability density of a normally distributed random variable.

In 2 Dimension, the Gaussian filter is given by,

$$g_\sigma(r,c) = \frac{1}{2\pi\sigma^2} e^{-(r^2+c^2)/2\sigma^2}$$

$$= \frac{1}{\sqrt{2\pi}\,\sigma} e^{-r^2/2\sigma^2} \frac{1}{\sqrt{2\pi}\,\sigma} e^{-c^2/2\sigma^2}$$

$$= g_\sigma(r)\, g_\sigma(c)$$

Hence the Gaussian filter is separable.

∴ it can be computed very efficiently

fig (b) Also gives a qualitative impression of the Freq^n response of the Gaussian filter

✓ It can be see that the Gaussian filter suppresses high freq^n much better than the mean filter.

Image transformation :-

Simple strategy → All the pixel in the i/p image, to transform their co-ordinate & to set the gray value of the transformed point in the O/P image.

But Unfortunately idea does not work.

eg:- An image is scaled by a factor of 2 : only one quarter of the pixel in the o/p image would be set.

correct way → All pixel in the o/p image & to calculate position of the corresponding pt. in the i/p image. (set pt of gray value)

1. **Nearest - Neighbor interpolation :-** ← several way.

The transformed pixel center lies on a non-integer position between four adjacent pixel centers.

Defn → The simplest & fastest interpolation method is to calculate the closest of the four adjacent pixel centers, which only involve rounding the floating pt co-ordinate of the transformed pixel centre & to use gray value of the closest pixel in i/p image. as the gray value of the pixel in O/P image. shown in fig 2(b) This interpolation method is called nearest - neighbor interpolation.

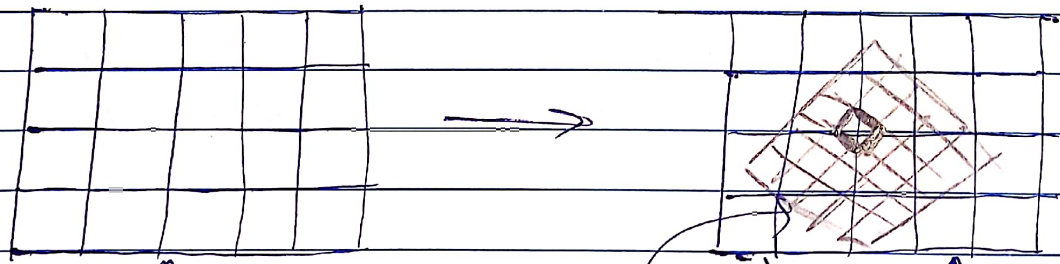


Fig 1. An affine Transformation of an image.

Note that integer co-ordinates in o/p image transform to non integer co-ordinate in the original image & hence must be interpolated.

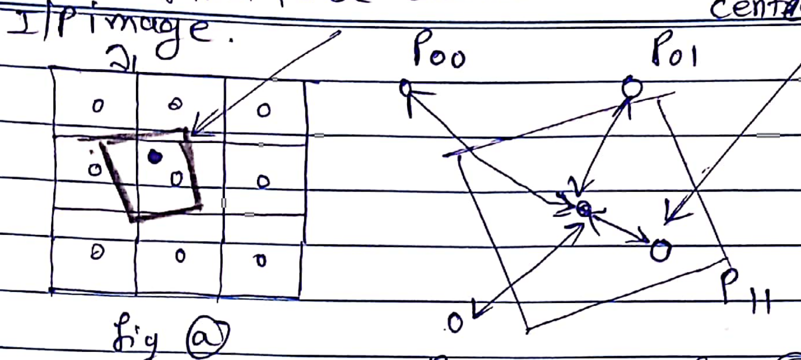


fig (a) fig (b)

Distance to the pixel centers in the i/p image.

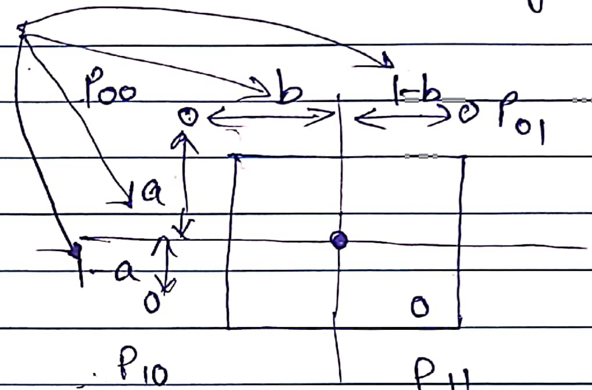


fig (c)

(a) A pixel in o/p image is transformed back to the i/p image. Note that the transformed pixel center lies at a non-integer position between four adjacent pixel centers.

(b) Nearest - neighbor interpolation determines the closest pixel center in the i/p image & uses its gray value in the o/p image.

(c) Bilinear interpolation determines the distances to the four adjacent pixel centers & weights their gray values using the distances.

★ Bilinear interpolation :-

AH 775324V
(a)

AH
(b) ← detail of a

✓ Image showing a serial number of a bank note.

AH 775324V
(c)

AH
(d) ← detail of c

✓ Image rotated such that the serial number is horizontal using nearest - neighbor interpolation.

AH775324V
(e)

AH
(f) ← detail of e

Image rotated using bilinear interpolation
(e) Note the smooth edges of the characters.

✓ If image is scaled by a factor > 1
To get a better interpolation, use more infm than the grey value of the closest pixel.

* Smoothing to Avoid Aliasing :-

✓ In Bilinear interpolation, interpolate from the closest four pixel centers. However, if image is scaled down adjacent pixel centers in the o/p image will not ne cessarily be close in the i/p image.

AH 775327 V		AH

(a) (b) ← details of (a)

Image showing a serial number of a bank note

AH 775327 V		AH

(c) (d) ← details of (c)

The image of scaled down by a factor of 3 using bilinear interpolation.

Note the different stroke widths of the vertical strokes of the letter H. This is caused by aliasing

AH 775327 V		AH

(e) (f) ← details of (e)

Result of scaling the image down by integrating a smoothing filter (mean filter) into image transformation.