**Digital image transformations:** one important transformation in images is 2D unitary transformations. They are utilized to extract features from images. An example is the Fourier transform where the average value or *dc term* is proportional to the average image amplitude, and the high frequency terms (*ac term*) give indication of the amplitude and the orientation of edges within the image. Dimensionality reduction is another image processing application. Stated simply, those transform coefficients that are small may be excluded from processing operations, such as filtering, without much loss in processing accuracy. Another application in the field of image coding is transform image coding, in which a bandwidth reduction is achieved by discarding or grossly quantizing low-magnitude transform coefficients.

**Fourier Transform & Shannon’s Theorem (Fourier Transform):** is an important image processing tool which is used to decompose an image into its sine and cosine components. The output of the transformation represents the image in the Fourier or frequency domain, while the input image is the spatial domain equivalent. In the Fourier domain image, each point represents a particular frequency contained in the spatial domain image. The Fourier Transform is used in a wide range of applications, such as image analysis, image filtering, image reconstruction and image compression.

In many cases in digital images it is used the discrete Fourier transform (DFT) which doesn’t contain all frequencies, but only a set of samples which is large enough to fully describe the spatial domain image. For an image of size NxN the DFT is given by this equation:



where f(a,b) is the image in the spatial domain and the exponential term is the basis function corresponding to each point F(k,l) in the Fourier space. The equation can be interpreted as: the value of each point F(k,l) is obtained by multiplying the spatial image with the corresponding base function and summing the result. In a similar way, the Fourier image can be re-transformed to the spatial domain. The inverse Fourier transform is given by:



To obtain the result for the above equations, a double sum has to be calculated for each image point. However, because the Fourier Transform is *separable*, it can be written as



where



Using these two formulas, the spatial domain image is first transformed into an intermediate image using *N* one-dimensional Fourier Transforms. This intermediate image is then transformed into the final image, again using *N* one-dimensional Fourier Transforms. Expressing the two-dimensional Fourier Transform in terms of a series of *2N* one-dimensional transforms decreases the number of required computations. Even with these computational savings, the ordinary one-dimensional DFT has complexity. This can be reduced to if we employ the *Fast Fourier Transform* (FFT) to compute the one-dimensional DFTs. This is a significant improvement, in particular for large images. There are various forms of the FFT and most of them restrict the size of the input image that may be transformed, often to where *n* is an integer. The mathematical details are well described in the literature. The Fourier Transform produces a complex number valued output image which can be displayed with two images, either with the *real* and *imaginary* part or with *magnitude* and *phase*. In image processing, often only the magnitude of the Fourier Transform is displayed, as it contains most of the information of the geometric structure of the spatial domain image. However, if we want to re-transform the Fourier image into the correct spatial domain after some processing in the frequency domain, we must make sure to preserve both magnitude and phase of the Fourier image. The Fourier domain image has a much greater range than the image in the spatial domain. Hence, to be sufficiently accurate, its values are usually calculated and stored in float values.

**Fourier Transform & Shannon’s Theorem (Shannon’s Theorem):** the area of interest is transmitting messages over a noisy channel. The definition is:

**Definition:** The input to ***a binary symmetric channel*** with parameter p is a sequence of bits x1, x2, . . . , and the output is a sequence of bits y1, y2, . . . , such that **Pr**|xi = yi|= 1−p independently for each i.

This means that every bit transmitted have the same probability to be flipped by the channel. The question is how much information we can send on the channel with this level of noise. Naturally, a channel would have some capacity constraints, and the question is how to send the largest amount of information, so that the receiver can recover the original information sent.

**Definition:** A (k, n) ***encoding function*** takes as input a sequence of *k* bits and outputs a sequence of *n* bits. A (k, n) decoding function takes as input a sequence of *n* bits and outputs a sequence of *k* bits.

Thus, the sender would use the encoding function to send its message, and the decoder would use the received string (with the noise in it), to recover the sent message. Thus, the sender starts with a message with k bits, it blow it up to n bits, using the encoding function, to get some robustness to noise, it send it over the (noisy) channel to the receiver. The receiver, takes the given (noisy) message with n bits, and uses the decoding function to recover the original k bits of the message. Naturally, we would like k to be as large as possible (for a fixed n), so that we can send as much information as possible on the channel.

The Shannon’s Theorem is defined:

For a binary symmetric channel with parameter p < ½ and for any constants δ, γ > 0, where n is sufficiently large, the following holds:

(i) For an *k ≤ n(1 − H(p) − δ)* there exists *(k, n)* encoding and decoding functions such that the probability the receiver fails to obtain the correct message is at most γ for every possible k-bit input messages.

(ii) There are no *(k, n)* encoding and decoding functions with *k ≥ n(1 − H(p) + δ)* such that the probability of decoding correctly is at least δ for a k-bit input message chosen uniformly at random.

**Image Noise:** An image may be subject to noise and interference from several sources, including electrical sensor noise, photographic grain noise, and channel errors. Noise can be cleaned using statistical methods and also *ad hoc* methods. Image noise arising from a noisy sensor or channel transmission errors usually appears as discrete isolated pixel variations that are not spatially correlated. Pixels that are in error often appear visually to be markedly different from their neighbors.

**Linear noise cleaning:** Noise added to an image generally has a higher-spatial-frequency spectrum than the normal image components because of its spatial decorrelatedness. Hence, simple low-pass filtering can be effective for noise cleaning.

***Spatial domain processing:*** a spatially filtered output image G(j,k) can be formed by discrete convolution of an input image F(j,k) with a LxL impulse response array according to the relation G(j,k) , where C = (L+1)/2.

For noise cleaning H should be of low-pass form with all positive elements.



These arrays, called *noise cleaning masks*, are normalized to unit weighting so that the noise-cleaning process does not introduce an amplitude bias in the processed image.

***Fourier Domain Processing:*** High-frequency noise effects can be reduced by Fourier domain filtering with a zonal low-pass filter. The sharp cutoff characteristic of the zonal low-pass filter leads to ringing artifacts in a filtered image. This deleterious effect can be eliminated by the use of a smooth cutoff filter, such as the Butterworth low-pass filter. Unlike convolution, Fourier domain processing, often provides quantitative and intuitive insight into the nature of the noise process, which is useful in designing noise cleaning spatial filters.

***Homomorphic Filtering:*** Homomorphic filtering (14) is a useful technique for image enhancement when an image is subject to multiplicative noise or interference. The input image F(j,k) is assumed to be modeled as the product of a noise-free image S(j, k) and an illumination interference array I(j,k). Thus: 

***Nonlinear noise cleaning:*** The linear processing techniques described previously perform reasonably well on images with continuous noise, such as additive uniform or Gaussian distributed noise. However, they tend to provide too much smoothing for impulse like noise. Nonlinear techniques often provide a better trade-off between noise smoothing and the retention of fine image detail.

***Outlier.*** is anoise cleaning technique in which each pixel is compared to the average of its eight neighbors. If the magnitude of the difference is greater than some threshold level, the pixel is judged to be noisy, and it is replaced by its neighborhood average. The eight-neighbor average can be computed by convolution of the observed image with the impulse response array.



***Median Filter.*** *Median filtering* is a nonlinear signal processing technique that is useful for noise suppression in images. In one-dimensional form, the median filter consists of a sliding window encompassing an odd number of pixels. The center pixel in the window is replaced by the median of the pixels in the window. The median of a discrete sequence *a*1, *a*2,..., *aN* for *N* odd is that member of the sequence for which (*N* – 1)/2 elements are smaller or equal in value and (*N* – 1)/2 elements are larger or equal in value. Operation of the median filter can be analyzed to a limited extent. It can be shown that the median of the product of a constant *K* and a sequence f(j) is: 

However, for two arbitrary sequences f(j) and g(j), it does not follow that the median of the sum of the sequences is equal to the sum of their medians. That is, in general: 

The concept of the median filter can be extended easily to two dimensions by utilizing a two-dimensional window of some desired shape such as a rectangle or discrete approximation to a circle. It is obvious that a two-dimensional LxL median filter will provide a greater degree of noise suppression than sequential processing with Lx1 median filters, but two-dimensional processing also results in greater signal suppression.

***Pseudomedian Filter:*** posses many of the properties as the median filter. Let {*SL*} denote a sequence of elements *s*1, *s*2,..., *sL*. The pseudomedian of the sequence is where for M = (L + 1)/2



Operationally, the sequence of *L* elements is decomposed into subsequences of *M* elements, each of which is slid to the right by one element in relation to its predecessor, and the appropriate MAX and MIN operations are computed.

***Wavelet De-noising.*** The usefulness of wavelet transforms for image coding derives from the property that most of the energy of a transformed image is concentrated in the trend transform components rather than the fluctuation components. The fluctuation components may be grossly quantized without serious image degradation. This energy compaction property can also be exploited for noise removal. The wavelet transform coefficients are thresholded such that the presumably noisy, low-amplitude coefficients are set to zero.

**Discrete Cosine (Fourier) transforms:** is a linear integral transformation similar to discrete Fourier transforms. In 1D cosines with growing frequencies constitute the basis function used for function expansion: it is a linear combination of these cosines and real numbers suffice for such an expansion. The DCT expansion corresponds to the double length of the DFT operating on a function with an even symmetry. Similarly to the DFT, the DCT operates on function samples of finite length, and a periodic extension of this function is needed to be able to perform DCT (or DFT) expansion. Consider the general case which covers both the discrete cosine transform (with even symmetry) and the discrete sine transform (with odd symmetry). The first choice has to be made about the symmetry at both left and right bounds of the signal, i.e., 2x2=4 possible options. The second choice is about which point the extension is performed, also at both left and right bounds of the signal, i.e., an additional 2x2=4 possible options. Altogether 4 x 4=16 possibilities are obtained. If we do not allow odd periodic extensions then the sine transforms are ruled out and 8 possible choices remain yielding 8 different types of DCT. If the same type of point is used for extension at left and right bounds then only half the options remain, i.e., 8/2=4. This yields four basic types of DCT—they are usually denoted by suffixing Roman numbers as DCT-I, DCT-II, DCT-III, DCT-IV.

The DCT can easily be generalized to two dimensions which is shown here for the square image, M = N. The 2D DCT-II is:

