SONA COLLEGE OF TECHNOLOGY

An endearing shrine for excellence in education

| An Autonomous Institution |

Department of Electronics and Communication Engineering

THE INSTITUTION OF ELECTRONICS AND TELECOMMUNICATION ENGINEERS

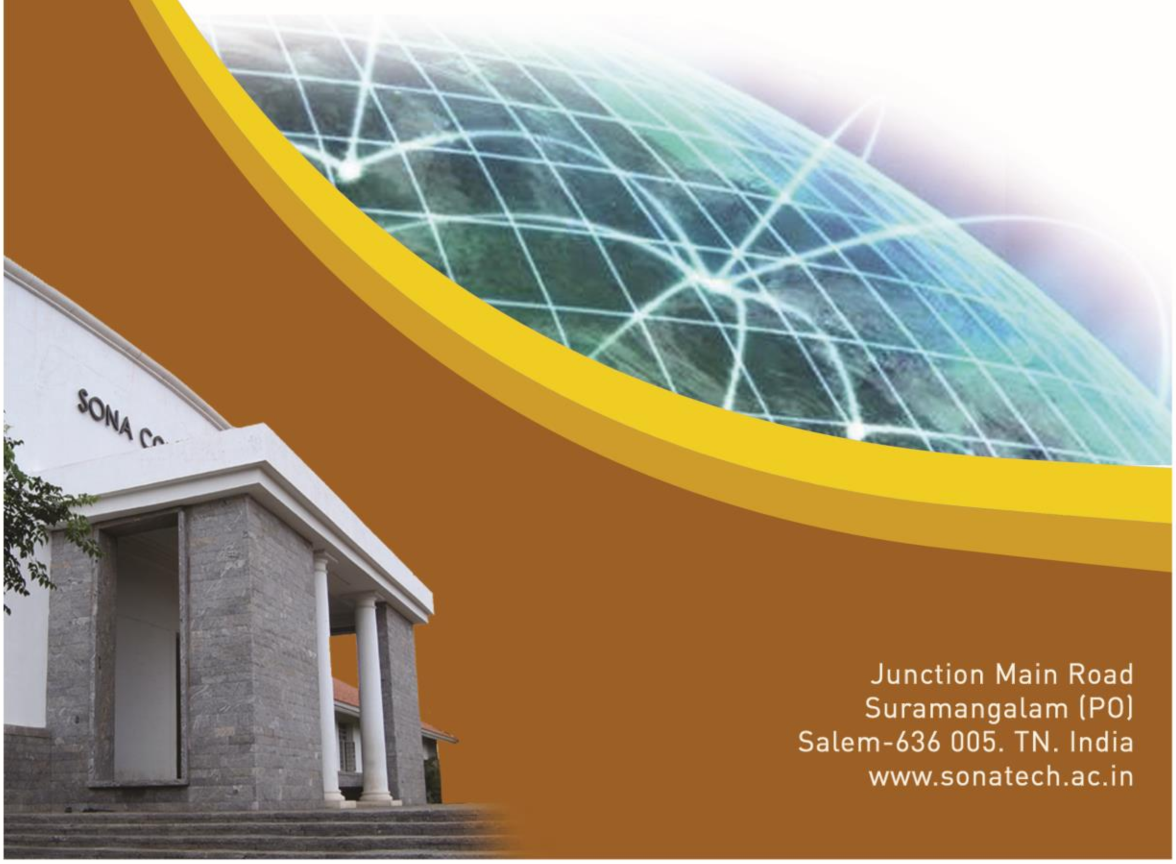
IETE

सह वीर्य करवावहै

K.J.S.I.E.I.T.

TRANSFORMS AND ALGORITHM IN SIGNAL AND IMAGE PROCESSING

2016-17



Junction Main Road

Suramangalam (PO)

Salem-636 005. TN. India

www.sonatech.ac.in

EDITOR'S BOARD

Editorial Head

Dr.R.S.Sabeenian,
Professor &Head, Dept of ECE,
Head R&D Sona SIPRO.

Editorial members:

1. Dr.K.R.Kavitha,
   Associate Professor

2. Ms.M.Jamuna Rani,
   Assist.Professor(Sr.G)

3. Ms.T.Shanthi,
   Assist.Professor(Sr.G)

Magazine Coordinator

M.Senthil Vadivu,
Assistant Professor

Student Members:

1. Dhatchaini.S

2. Divya Bharathy.J

Preface

The field of signal and image processing encompasses the theory and practice of algorithms and hardware that convert signals produced by artificial or natural means into a form useful for a specific purpose. The signals might be speech, audio, images, video, sensor data, telemetry, electrocardiograms, or seismic data, among others; possible purposes include transmission, display, storage, interpretation, classification, segmentation, or diagnosis.

Current research in digital signal processing includes robust and low complexity filter design, signal reconstruction, filter bank theory, and wavelets. In statistical signal processing, the areas of research include adaptive filtering, learning algorithms for neural networks, spectrum estimation and modeling, and sensor array processing with applications in sonar and radar. Image processing work is in restoration, compression, quality evaluation, computer vision, and medical imaging. Speech processing research includes modeling, compression, and recognition. Video compression, analysis, and processing projects include error concealment technique for 3D compressed video, automated and distributed crowd analytics, stereo-to-auto stereoscopic 3D video conversion, virtual and augmented reality.

Techniques in digital image processing

Pharkavi.M, II Year

Digital image processing is the use of computer algorithms to perform image processing on digital images. As a subcategory or field of digital signal processing, digital image processing has many advantages over analog image processing. It allows a much wider range of algorithms to be applied to the input data and can avoid problems such as the build-up of noise and signal distortion during processing. Since images are defined over two dimensions digital image processing may be modeled in the form of multidimensional systems.

Many of the techniques of digital image processing, or digital picture processing as it often was called, were developed in the 1960s at the Jet Propulsion Laboratory, Massachusetts Institute of Technology, Bell Laboratories, University of Maryland, and a few other research facilities, with application to satellite imagery, wire-photo standards conversion, medical imaging, videophone, character recognition, and photograph enhancement. The cost of processing was fairly high, however, with the computing equipment of that era. That changed in the 1970s, when digital image processing proliferated as cheaper computers and dedicated hardware became available. Images then could be processed in real time, for some dedicated problems such as television standards conversion. As general-purpose computers became faster, they started to take over the role of dedicated hardware for all but the most specialized and computer-intensive operations. With the fast computers and signal processors available in the 2000s, digital image processing has become the most common form of image processing and generally, is used because it is not only the most versatile method, but also the cheapest.

Digital image processing technology for medical applications was inducted into the Space Foundation Space Technology Hall of Fame in 1994.

Feature extraction:

Feature extraction starts from an initial set of measured data and builds derived values intended to be informative and non-redundant, facilitating the subsequent learning and generalization steps, and in some cases leading to better human interpretations. Feature extraction is related to dimensionality reduction.

When the input data to an algorithm is too large to be processed and it is suspected to be redundant, then it can be transformed into a reduced set of features .Determining a subset of the initial features is called *feature selection*. The selected features are expected to contain the relevant information from the input data, so that the desired task can be performed by using this reduced representation instead of the complete initial data.

Feature extraction involves reducing the amount of resources required to describe a large set of data. When performing analysis of complex data one of the major problems stems from the number of variables involved. Analysis with a large number of variables generally requires a large amount of memory and computation power, also it may cause a classification algorithm to over fit to training samples and generalize poorly to new samples. Feature extraction is a general term for methods of constructing combinations of the variables to get around these problems while still describing the data with sufficient accuracy. Many machine learning practitioners believe that properly optimized feature extraction is the key to effective model construction.

Time and frequency analysis in image processing

Padmasaranya.V, II Year

Signal processing concerns the analysis, synthesis, and modification of signals, which are broadly defined as functions conveying "information about the behavior or attributes of some phenomenon", such as sound, images, and biological measurements. For example, signal processing techniques are used to improve signal transmission fidelity, storage efficiency, and subjective quality, and to emphasize or detect components of interest in a measured signal.

Digital signal processing (DSP) is the use of digital processing, such as by computers or more specialized digital signal processors, to perform a wide variety of signal processing operations. The signals processed in this manner are a sequence of numbers that represent samples of a continuous variable in a domain such as time, space, or frequency.

Digital signal processing and analog signal processing are subfields of signal processing. DSP applications include audio and speech processing, sonar, radar and other sensor array processing, spectral density estimation, statistical signal processing, digital image processing, signal processing for telecommunications, control systems, biomedical engineering, seismology, among others.

DSP can involve linear or nonlinear operations. Nonlinear signal processing is closely related to nonlinear system identification and can be implemented in the time, frequency, and spatiotemporal domains.

The application of digital computation to signal processing allows for many advantages over analog processing in many applications, such as error detection and correction in transmission as well as data compression. DSP is applicable to both streaming data and static data.

To digitally analyze and manipulate an analog signal, it must be digitized with an analog-to-digital converter (ADC). Sampling is usually carried out in two stages, discretization and quantization. Discretization means that the signal is divided into equal intervals of time, and each interval is represented by a single measurement of amplitude. Quantization means each amplitude measurement is approximated by a value from a finite set. Rounding real numbers to integers is an example.

The Nyquist–Shannon sampling theorem states that a signal can be exactly reconstructed from its samples if the sampling frequency is greater than twice the highest frequency component in the signal. In practice, the sampling frequency is often significantly higher than twice the Nyquist frequency.

Theoretical DSP analyses and derivations are typically performed on discrete-time signal models with no amplitude inaccuracies created by the abstract process of sampling. Numerical methods require a quantized signal, such as those produced by an ADC. The processed result might be a frequency spectrum or a set of statistics. But often it is another quantized signal that is converted back to analog form by a digital-to-analog converter (DAC).

In DSP, engineers usually study digital signals in one of the following domains: time domain spatial domain frequency domain, and wavelet domains. They choose the domain in which to process a signal by making an informed assumption as to which domain best represents the essential characteristics of the signal and the processing to be applied to it. A sequence of samples from a measuring device produces a temporal or spatial domain representation, whereas a discrete Fourier transform produces the frequency domain representation.

The most common processing approach in the time or space domain is enhancement of the input signal through a method called filtering. Digital filtering generally consists of some linear transformation of a number of surrounding samples around the current sample of the input or output signal. There are various ways to characterize filters

- A *linear* filter is a linear transformation of input samples; other filters are *nonlinear*. Linear filters satisfy the superposition principle. if an input is a weighted linear combination of different signals, the output is a similarly weighted linear combination of the corresponding output signals.
- A *causal* filter uses only previous samples of the input or output signals; while a *non-causal* filter uses future input samples. A non-causal filter can usually be changed into a causal filter by adding a delay to it.
- A time-invariant filter has constant properties over time; other filters such as adaptive filters change in time.
- A stable filter produces an output that converges to a constant value with time or remains bounded within a finite interval. An unstable filter can produce an output that grows without bounds, with bounded or even zero input.
- A finite impulse response filter uses only the input signals, while an infinite impulse response filter uses both the input signal and previous samples of the output signal. FIR filters are always stable, while IIR filters may be unstable.

A filter can be represented by a block diagram, which can then be used to derive a sample processing algorithm to implement the filter with hardware instructions. A filter may also be described as a difference equation, a collection of zeros and poles or an impulse response or step response.

The output of a linear digital filter to any given input may be calculated by convolving the input signal with the impulse response.

Frequency domain

Signals are converted from time or space domain to the frequency domain usually through use of the Fourier transform. The Fourier transform converts the time or space information to a magnitude and phase component of each frequency. With some applications, how the phase varies with frequency can be a significant consideration. Where phase is unimportant, often the Fourier transform is converted to the power spectrum, which is the magnitude of each frequency component squared.

The most common purpose for analysis of signals in the frequency domain is analysis of signal properties. The engineer can study the spectrum to determine which frequencies are present in the input signal and which are missing. Frequency domain analysis is also called *spectrum-* or *spectral analysis*.

Filtering, particularly in non-real time work can also be achieved in the frequency domain, applying the filter and then converting back to the time domain. This can be an efficient implementation and can give essentially any filter response including excellent approximations to brick wall filters.

There are some commonly-used frequency domain transformations. For example, the cestrum converts a signal to the frequency domain through Fourier transform, takes the logarithm, then applies another Fourier transform. This emphasizes the harmonic structure of the original spectrum.

Digital filters come in both IIR and FIR types. FIR filters have many advantages but are computationally more demanding. Whereas FIR filters are always stable, IIR filters have feedback loops that may resonate when stimulated with certain input signals. The Z-transform provides a tool for analyzing potential stability issues of digital IIR filters. It is analogous to the Laplace transform, which is used to design analog IIR filters.

In numerical analysis and functional analysis, a discrete wavelet transform is any wavelet transform for which the wavelets are discretely sampled. As with other wavelet transforms, a key advantage it has over Fourier transforms is temporal resolution: it captures both frequency *and* location information. The accuracy of the joint time-frequency resolution is limited by the uncertainty principle of time-frequency.

Pattern recognition in image processing

Sanchna.T, II Year

Pattern recognition is a branch of machine learning that focuses on the recognition of patterns and regularities in data, although it is in some cases considered to be nearly synonymous with machine learning. Pattern recognition systems are in many cases trained from labeled training data but when no labeled data are available other algorithms can be used to discover previously unknown patterns.

The terms pattern recognition, machine learning, data mining and knowledge discovery in databases are hard to separate, as they largely overlap in their scope. Machine learning is the common term for supervised learning methods and originates from artificial intelligence, whereas KDD and data mining have a larger focus on unsupervised methods and stronger connection to business use. Pattern recognition has its origins in engineering, and the term is popular in the context of computer vision: a leading computer vision conference is named Conference on Computer Vision and Pattern Recognition. In pattern recognition, there may be a higher interest to formalize, explain and visualize the pattern, while machine learning traditionally focuses on maximizing the recognition rates. Yet, all of these domains have evolved substantially from their roots in artificial intelligence, engineering and statistics, and they've become increasingly similar by integrating developments and ideas from each other.

In machine learning, pattern recognition is the assignment of a label to a given input value. In statistics, discriminant analysis was introduced for this same purpose in 1936. An example of pattern recognition is classification, which attempts to assign each input value to one of a given set of *classes*. However, pattern recognition is a more general problem that encompasses other types of output as well. Other examples are regression, which assigns a real-valued output to each input; sequence labeling, which assigns a class to each member of a sequence of values and parsing, which assigns a parse tree to an input sentence, describing the syntactic structure of the sentence.

Pattern recognition algorithms:

Pattern recognition algorithms generally aim to provide a reasonable answer for all possible inputs and to perform most likely matching of the inputs, considering their statistical variation. This is opposed to *pattern matching* algorithms, which look for exact matches in the input with pre-existing patterns. A common example of a pattern-matching algorithm is regular expression matching, which looks for patterns of a given sort in textual data and is included in the search capabilities of many text editors and word processors. In contrast to pattern recognition, pattern matching is not generally a type of machine learning, although pattern-matching algorithms can sometimes succeed in providing similar-quality output of the sort provided by pattern-recognition algorithms.

Pattern recognition is generally categorized according to the type of learning procedure used to generate the output value. Supervised learning assumes that a set of training data has been provided, consisting of a set of instances that have been properly labeled by hand with the correct output. A learning procedure then generates a model that attempts to meet two sometimes conflicting objectives: Perform as well as possible on the training data and generalize as well as possible to new. Unsupervised learning, on the other hand, assumes training data that has not been hand-labeled, and attempts to find inherent patterns in the data that can then be used to determine the correct output value for new data instances. A combination of the two that has recently been explored is semi-supervised learning, which uses a combination of labeled and unlabeled. Note that in cases of unsupervised learning, there may be no training data at all to speak of; in other words, and the data to be labeled is the training data.

Sometimes different terms are used to describe the corresponding supervised and unsupervised learning procedures for the same type of output. The unsupervised equivalent of classification is normally known as clustering, based on the common perception of the task as involving no training data to speak of, and of grouping the input data into clusters based on some inherent similarity measure rather than assigning each input instance into one of a set of pre-defined classes. Note also that in some fields, the terminology is different: For example, in community ecology, the term "classification" is used to refer to what is commonly known as clustering.

The piece of input data for which an output value is generated is formally termed an instance. The instance is formally described by a vector of features, which together constitute a description of all known characteristics of the instance. These feature vectors can be seen as defining points in an appropriate multidimensional space, and methods for manipulating vectors in vector spaces can be correspondingly applied to them, such as computing the dot product or the angle between two vectors. Typically, features are either categorical also known as nominal, i.e., consisting of one of a set of unordered items, such as a gender of "male" or "female", or a blood type of "A", "B", "AB" or "O", ordinal consisting of one of a set of ordered items, e.g., "large", "medium" or "small", integer-valued e.g., a count of the number of occurrences of a particular word in an email or real-valued e.g., a measurement of blood pressure Often, categorical and ordinal data are grouped together; likewise for integer-valued and real-valued data. Furthermore, many algorithms work only in terms of categorical data and require that real-valued or integer-valued data be *discretized* into groups.

Many common pattern recognition algorithms are *probabilistic* in nature, in that they use statistical inference to find the best label for a given instance. Unlike other algorithms, which simply output a "best" label, often probabilistic algorithms also output a probability of the instance being described by the given label. In addition, many probabilistic algorithms output a list of the *N*-best labels with associated probabilities, for some value of *N*, instead of simply a single best label. When the number of possible labels is fairly small (e.g., in the case of classification), *N* may be set so that the probability of all possible labels is output. Probabilistic algorithms have many advantages over non-probabilistic algorithms:

- They output a confidence value associated with their choice. Some other algorithms may also output confidence values, but in general, only for probabilistic algorithms is this value mathematically grounded in probability theory. Non-probabilistic confidence values can in general not be given any specific meaning, and only used to compare against other confidence values output by the same algorithm.
- Correspondingly, they can *abstain* when the confidence of choosing any particular output is too low.
- Because of the probabilities output, probabilistic pattern-recognition algorithms can be more effectively incorporated into larger machine-learning tasks, in a way that partially or completely avoids the problem of error propagation.

Wavelet transform

Monisha.A, II Year

A **wavelet** is a wave-like oscillation with an amplitude that begins at zero, increases, and then decreases back to zero. It can typically be visualized as a "brief oscillation" like one recorded by a seismograph or heart monitor. Generally, wavelets are intentionally crafted to have specific properties that make them useful for signal processing. Using a "reverse, shift, multiply and integrate" technique called convolution, wavelets can be combined with known portions of a damaged signal to extract information from the unknown portions.

A wavelet could be created to have a frequency of Middle C and a short duration of roughly a 32nd note. If this wavelet were to be convolved with a signal created from the recording of a song, then the resulting signal would be useful for determining when the Middle C note was being played in the song. Mathematically, the wavelet will correlate with the signal if the unknown signal contains information of similar frequency. This concept of correlation is at the core of many practical applications of wavelet theory.

As a mathematical tool, wavelets can be used to extract information from many kinds of data, including – but certainly not limited to – audio signals and images. Sets of wavelets are generally needed to analyze data fully. A set of "complementary" wavelets will decompose data without gaps or overlap so that the decomposition process is mathematically reversible. Thus, sets of complementary wavelets are useful in wavelet based compression/decompression algorithms where it is desirable to recover the original information with minimal loss.

In formal terms, this representation is a wavelet series representation of a square-integral function with respect to either a complete, orthonormal set of basic functions, or an over complete set or frame of a vector space, for the Hilbert space of square integral functions. This is accomplished through coherent states.

In any discretized wavelet transform, there are only a finite number of wavelet coefficients for each bounded rectangular region in the upper half plane. Still, each coefficient requires the evaluation of an integral. In special situations this numerical complexity can be avoided if the scaled and shifted wavelets form a multiresolution analysis. This means that there has to exist an auxiliary function, the *father wavelet* ϕ in $L^2(\mathbf{R})$, and that is an integer. A typical choice is $a = 2$ and $b = 1$. The most famous pair of father and mother wavelets is the Daubechies 4-tap wavelet. Note that not every orthonormal discrete wavelet basis can be associated to a multiresolution analysis; for example, the Journee wavelet admits no multiresolution analysis.

Multiplication with a rectangular window in the time domain corresponds to convolution with a function in the frequency domain, resulting in spurious ringing artifacts for short/localized temporal windows. With the Continuous-time Fourier Transform and this convolution is with a delta function in Fourier space, resulting in the true Fourier transform of the signal. The window function may be some other apodizing filter, such as a Gaussian. The choice of windowing function will affect the approximation error relative to the true Fourier transform.

A given resolution cell's time-bandwidth product may not be exceeded with the STFT. All STFT basis elements maintain a uniform spectral and temporal support for all temporal shifts or offsets, thereby attaining an equal resolution in time for lower and higher frequencies. The resolution is purely determined by the sampling width.

In contrast, the wavelet transform's multiresolution properties enables large temporal supports for lower frequencies while maintaining short temporal widths for higher frequencies by the scaling properties of the wavelet transform. This property extends conventional time-frequency analysis into time-scale analysis.

STFT time-frequency atoms (left) and DWT time-scale atoms (right). The time-frequency atoms are four different basis functions used for the STFT. The time-scale atoms of the DWT achieve small temporal widths for high frequencies and good temporal widths for low frequencies with a single transform basis set.

The discrete wavelet transform is less computationally complex, taking O(N) time as compared to O($N \log N$) for the fast Fourier transform. This computational advantage is not inherent to the transform but reflects the choice of a logarithmic division of frequency, in contrast to the equally spaced frequency divisions of the FFT which uses the same basis functions as DFT (Discrete Fourier Transform). It is also important to note that this complexity only applies when the filter size has no relation to the signal size. A wavelet without compact support such as the Shannon wavelet would require O (N^2). (For instance, a logarithmic Fourier Transform also exists with O(N) complexity, but the original signal must be sampled logarithmically in time, which is only useful for certain types of signals.

Scaling filter

An orthogonal wavelet is entirely defined by the scaling filter – a low-pass finite impulse response (FIR) filter of length 2N and sum 1. In biorthogonal wavelets, separate decomposition and reconstruction filters are defined.

For analysis with orthogonal wavelets the high pass filter is calculated as the quadrature mirror filter of the low pass, and reconstruction filters are the time reverse of the decomposition filters.

Daubechies and Symlet wavelets can be defined by the scaling filter.

Scaling function

Wavelets are defined by the wavelet function ψ (*t*) (i.e. the mother wavelet) and scaling function ϕ (*t*) (also called father wavelet) in the time domain.

The wavelet function is in effect a band-pass filter and scaling it for each level halves its bandwidth. This creates the problem that to cover the entire spectrum, an infinite number of levels would be required. The scaling function filters the lowest level of the transform and ensures all the spectrum is covered. See for a detailed explanation.

For a wavelet with compact support, ϕ (*t*) can be considered finite in length and is equivalent to the scaling filter *g*.

Wavelet function

The wavelet only has a time domain representation as the wavelet function ψ (*t*).

For instance, Mexican hat wavelets can be defined by a wavelet function. See a list of a few Continuous wavelets.

Singular value decomposition

Priyanka, II Year

Singular Value Decomposition (SVD) has recently emerged as a new paradigm for processing different types of images. SVD is an attractive algebraic transform for image processing applications. The paper proposes an experimental survey for the SVD as an efficient transform in image processing applications. Despite the well-known fact that SVD offers attractive properties in imaging, the exploring of using its properties in various image applications is currently at its infancy. Since the SVD has many attractive properties have not been utilized, this paper contributes in using these generous properties in newly image applications and gives a highly recommendation for more research challenges. In this paper, the SVD properties for images are experimentally presented to be utilized in developing new SVD-based image processing applications. The paper offers survey on the developed SVD based image applications. The paper also proposes some new contributions that were originated from SVD properties analysis in different image processing. The aim of this paper is to provide a better understanding of the SVD in image processing and identify important various applications and open research directions in this increasingly important area; SVD based image processing in the future research

The SVD is the optimal matrix decomposition in a least square sense that it packs the maximum signal energy into as few coefficients as possible. Singular value decomposition (SVD) is a stable and effective method to split the system into a set of linearly independent components, each of them bearing own energy contribution. Singular value decomposition (SVD) is a numerical technique used to diagonalizable matrices in numerical analysis. SVD is an attractive algebraic transform for image processing, because of its endless advantages, such as maximum energy packing which is usually used in compression. Ability to manipulate the image in base of two distinctive subspaces data and noise subspaces, which is usually uses in noise filtering and was utilized in watermarking applications. Each of these applications exploit key properties of the SVD. Also, it is usually used in solving of least squares problem, computing pseudo- inverse of a matrix and multivariate analysis. SVD is robust and reliable orthogonal matrix decomposition methods, which is due to its conceptual and stability reasons becoming more and more popular in signal processing area. SVD can adapt to the variations in local statistics of an image. Many SVD properties are attractive and are still not fully utilized. This paper provides thoroughly experiments for the generous properties of SVD that are not yet totally exploited in digital image processing. The developed SVD based image processing techniques were focused in compression, watermarking and quality measure. Experiments in this paper are performed to validate some of will know but unutilized properties of SVD in image processing applications. This paper contributes in utilizing SVD generous properties that are not unexploited in image processing. This paper also introduces new trends and challenges in using SVD in image processing applications. Some of these new trends are well examined experimentally in this paper and validated and others are demonstrated and needs more work to be maturely validated. This paper opens many tracks for future work in using SVD as an imperative tool in signal processing. Organization of this paper is as follows. Section two introduces the SVD. Section three explores the SVD properties with their examining in image processing. Section four provides the SVD rank approximation and subspaces-based image applications. Section five explores SVD singular value-based image applications. Section six investigates SVD singular vectors-based image applications. Section seven provides SVD based image applications open issues and research trends.

SVD IMAGE PROPERTIES

Naveena.P, II Year

SVD is robust and reliable orthogonal matrix decomposition method. Due to SVD conceptual and stability reasons, it becomes more and more popular in signal processing area. SVD is an attractive algebraic transform for image processing. SVD has prominent properties in imaging. This section explores the main SVD properties that may be utilized in image processing. Although some SVD properties are fully utilized in image processing, others still need more investigation and contributed to. Several SVD properties are highly advantageous for images such as; its maximum energy packing, solving of least squares problem, computing pseudo- inverse of a matrix and multivariate analysis. A key property of SVD is its relation to the rank of a matrix and its ability to approximate matrices of a given rank. Digital images are often represented by low rank matrices and, therefore, able to be described by a sum of a relatively small set of Eigen images. This concept rises the manipulating of the signal as two distinct subspaces. Some hypotheses will be provided and verified in the following sections. For a complete review, the theoretical SVD related theorems are firstly summarized, and then the practical properties are reviewed associated with some experiments. SVD Subspaces: SVD is constituted from two orthogonal dominant and subdominant subspaces. This corresponds to partition the M-dimensional vector space into dominant and subdominant subspaces. This attractive property of SVD is utilized in noise filtering and watermarking. ▯ SVD architecture: For SVD decomposition of an image, singular value (SV) specifies the luminance of an image layer while the corresponding pair singular vectors (SCs) specify the geometry of the image layer. The largest object components in an image found using the SVD generally correspond to Eigen images associated with the largest singular values, while image noise corresponds to Eigen images associated with the SVs

PCA versus SVD: Principle component analysis (PCA) is also called the Karhunen-Loéve transform (KLT) or the hoteling transform. PCA is used to compute the dominant vectors representing a given data set and provides an optimal basis for minimum mean squared reconstruction of the given data. The computational basis of PCA is the calculation of the SVD of the data matrix, or equivalently the eigenvalues decomposition of the data covariance matrix SVD is closely related to the standard eigenvalues-eigenvector or spectral decomposition of a square matrix X, into VLV', where V is orthogonal, and L are diagonal. In fact, U and V of SVD represent the eigenvectors for XX' and X'X respectively. If X is symmetric, the singular values of X are the absolute value of the eigenvalues of X. SVD Multiresolution: SVD has the maximum energy packing among the other transforms. In many applications, it is useful to obtain a statistical characterization of an image at several resolutions. SVD decomposes a matrix into orthogonal components with which optimal sub rank approximations may be obtained. With the multiresolution SVD, the following important characteristics of an image may be measured, at each of the several levels of resolution: isotropy, spercity of principal components, self-similarity under scaling, and resolution of the mean squared error into meaningful components.SVD Oriented Energy: In SVD analysis of oriented energy both rank of the problem and signal space orientation can be determined. SVD is a stable and effective method to split the system into a set of linearly independent components, each of them bearing its own energy contribution. SVD is represented as a linear combination of its principle components, a few dominate components are bearing the rank of the observed system and can be severely reduced. The oriented energy concept is an effective tool to separate signals from different sources, or to select signal subspaces of maximal signal activity and integrity. Recall that the singular values represent the square root of the energy in corresponding principal direction. The dominant direction could equal to the first singular vector V1 from the SVD decomposition. Accuracy of dominance of the estimate could be measured by obtaining the difference or normalized difference between the first two SVs.Some of the SVD properties are not fully utilized in image processing applications. These unused properties will be experimentally conducted in the following sections for more convenient utilization of these properties in various images processing application. Much research work needs to be done in utilizing this generous transform.

Discrete cosine transforms in image processing

Nivetha.A, II Year

A **discrete cosine transform** (**DCT**) expresses a finite sequence of data points in terms of a sum of cosine functions oscillating at different frequencies. DCTs are important to numerous applications in science and engineering, from lossy compression of audio (e.g. MP3) and images (e.g. JPEG) (where small high-frequency components can be discarded), to spectral methods for the numerical solution of partial differential equations. The use of cosine rather than sine functions is critical for compression, since it turns out (as described below) that fewer cosine functions are needed to approximate a typical signal, whereas for differential equations the cosines express a particular choice of boundary conditions.

In particular, a DCT is a Fourier-related transform similar to the discrete Fourier transform (DFT), but using only real numbers. The DCTs are generally related to Fourier series coefficients of a periodically and symmetrically extended sequence whereas DFTs are related to Fourier series coefficients of a periodically extended sequence. DCTs are equivalent to DFTs of roughly twice the length, operating on real data with even symmetry (since the Fourier transform of a real and even function is real and even), whereas in some variants the input and/or output data are shifted by half a sample. There are eight standard DCT variants, of which four are common.

The most common variant of discrete cosine transform is the type-II DCT, which is often called simply "the DCT", Its inverse, the type-III DCT, is correspondingly often called simply "the inverse DCT" or "the IDCT". Two related transforms are the discrete sine transform (DST), which is equivalent to a DFT of real and *odd* functions, and the modified discrete cosine transform (MDCT), which is based on a DCT of *overlapping* data. Multidimensional DCTs (MD DCTs) are developed to extend the concept of DCT on MD Signals. There are several algorithms to compute MD DCT. A new variety of fast algorithms are also developed to reduce the computational complexity of implementing DCT.

The DCT, and the DCT-II, is often used in signal and image processing, especially for lossy compression, because it has a strong "energy compaction" property: in typical applications, most of the signal information tends to be concentrated in a few low-frequency components of the DCT. For strongly correlated Markov_processes, the DCT can approach the compaction efficiency of the Karhunen-Loève_transform (which is optimal in the decorrelation sense). As explained below, this stem from the boundary conditions implicit in the cosine functions.

DCT-II (bottom) compared to the DFT (middle) of an input signal (top).

A related transform, the modified_discrete_cosine_transform, or MDCT (based on the DCT-IV), is used in AAC, Verbs, WMA, and MP3 audio compression.DCTs are also widely employed in solving partial differential equations by spectral methods, where the different variants of the DCT correspond to slightly different even/odd boundary conditions at the two ends of the array.DCTs are also closely related to Chebyshev polynomials, and fast DCT algorithms (below) are used in Chebyshev_approximation of arbitrary functions by series of Chebyshev polynomials.

Multidimensional DCTs (MD DCTs) have several applications mainly 3-D DCT-II has several new applications like Hyper spectral Imaging coding systems, variable temporal length 3-D DCT coding, video_coding algorithms, adaptive video coding and 3-D Compression. Due to enhancement in the hardware, software and introduction of several fast algorithms, the necessity of using M-D DCTs is rapidly increasing. DCT-IV has gained popularity for its applications in fast implementation of real-valued polyphone filtering banks, lapped orthogonal transform and cosine-modulated wavelet bases. The DCT, and the DCT-II, is often used in signal and image processing, especially for lossy compression, because it has a strong "energy compaction" property: in typical applications, most of the signal information tends to be concentrated in a few low-frequency components of the DCT. For strongly correlated Markov processes, the DCT can approach the compaction efficiency of the Karhunen-Loève_transform (which is optimal in the decorrelation sense). As explained below, this stem from the boundary conditions implicit in the cosine functions. Like any Fourier-related transform, discrete cosine transforms (DCTs) express a function or a signal in terms of a sum of sinusoids with different frequencies and amplitudes. Like the discrete Fourier transform (DFT), a DCT operates on a function at a finite number of discrete data points. The obvious distinction between a DCT and a DFT is that the former uses only cosine functions, while the latter uses both cosines and sines (in the form of complex exponentials). However, this visible difference is merely a consequence of a deeper distinction: a DCT implies different boundary conditions from the DFT or other related transforms.

The Fourier-related transforms that operate on a function over a finite domain, such as the DFT or DCT or a Fourier series, can be thought of as implicitly defining an *extension* of that function outside the domain. Extension of the original function. A DCT, like a cosine transform, implies an even extension of the original function. However, because DCTs operate on *finite*, *discrete* sequences, two issues arise that do not apply for the continuous cosine transform. First, one has to specify whether the function is even or odd at *both* the left and right boundaries of the domain (i.e. the min-*n* and max-*n* boundaries in the definitions below, respectively). Second, one has to specify around *what point* the function is even or odd. Consider a sequence *abcd* of four equally spaced data points, and say that we specify an even *left* boundary. There are two sensible possibilities: either the data are even about the sample *a*, in which case the even extension is *dcbabcd*, or the data are even about the point *halfway* between *a* and the previous point, in which case the even extension is *dcbaabcd* (*a* is repeated). These different boundary conditions strongly affect the applications of the transform and lead to uniquely useful properties for the various DCT types. Most directly, when using Fourier-related transforms to solve partial differential equations by spectral methods, the boundary conditions are directly specified as a part of the problem being solved. Or, for the MDCT (based on the type-IV DCT), the boundary conditions are intimately involved in the MDCT's critical property of time-domain aliasing cancellation. In a subtler fashion, the boundary conditions are responsible for the "energy compactification" properties that make DCTs useful for image and audio compression, because the boundaries affect the rate of convergence of any Fourier-like series.

In particular, it is well known that any discontinuities in a function reduce the rate of convergence of the Fourier series, so that more sinusoids are needed to represent the function with a given accuracy. The same principle governs the usefulness of the DFT and other transforms for signal compression; the smoother a function is, the fewer terms in its DFT or DCT are required to represent it accurately, and the more it can be compressed. (Here, we think of the DFT or DCT as approximations for the Fourier series or cosine series of a function, respectively, in order to talk about its "smoothness".) However, the implicit periodicity of the DFT means that discontinuities usually occur at the boundaries: any random segment of a signal is unlikely to have the same value at both the left and right boundaries. (A similar problem arises for the DST, in which the odd left boundary condition implies a discontinuity for any function that does not happen to be zero at that boundary.) In contrast, a DCT where *both* boundaries are even *always* yields a continuous extension at the boundaries (although the slope is generally discontinuous). Therefore DCTs, and in particular DCTs of types I, II, V, and VI (the types that have two even boundaries) generally perform better for signal compression than DFTs and DSTs. In practice, a type-II DCT is usually preferred for such applications, in part for reasons of computational convenience.