DOKUZ EYLÜL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

CHARACTER AND OBJECT RECOGNITION BY USING IMAGE AND SOUND FUSION

by
Bertan KARAHODA

January, 2007
İZMİR

CHARACTER AND OBJECT RECOGNITION BY USING IMAGE AND SOUND FUSION

A Thesis Submitted to the
Graduate School of Natural and Applied Sciences of Dokuz Eylül University
In Partial Fulfillment of the Requirements for the Degree of Master of Science
in Mechanical Engineering, Machine Theory and Dynamics Program

by
Bertan KARAHODA

January, 2007
İZMİR

M.Sc THESIS EXAMINATION RESULT FORM

We have read the thesis entitled **"CHARACTER AND OBJECT RECOGNITION BY USING IMAGE AND SOUND FUSION"** completed by **Bertan KARAHODA** under supervision of **PROF. DR. EROL UYAR** and we certify that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Prof. Dr. Erol UYAR

Supervisor

Prof. Dr. Cüneyt GÜZELİŞ

(Jury Member)

Yrd. Doç. Dr. Zeki KIRAL

(Jury Member)

Prof.Dr. Cahit HELVACI
Director
Graduate School of Natural and Applied Sciences

ACKNOWLEDGEMENTS

I am greatly indebted to my thesis supervisor Prof. Dr. Erol UYAR for his help, kind interest and encouragement throughout the development of this study.

I would also like to thank to my family for their moral support and my friend Tayfun OMER for helping me with the collection of experimental data.

Bertan KARAHODA

CHARACTER AND OBJECT RECOGNITION BY USING IMAGE AND SOUND FUSION

ABSTRACT

In this thesis, a novel approach that integrates image and sound recognition tasks is presented. Human brain is too complex to be analyzed. But some of the main learning tasks in the brain can be simulated. The learning processes in the brain cannot be considered in the absence of sound or image. Both image and sound plays very important role in the learning processes. Therefore in this thesis the new algorithms that integrate the sound and image recognition tasks are tried in order to become closer to the operation of the human brain. The algorithm takes the sound and image input simultaneously, and extracts the meaningful features and then integrates them as a new knowledge in the data field. After the new knowledge is saved then the new image or sound inputs corresponding to the same knowledge are used to teach the algorithm. The user takes role as a teacher in the algorithm. The experimental results show that the proposed approach is applicable on the learning processes.

Keywords : Speech recognition, image recognition, image and sound fusion, artificial intelligence.

SES VE GÖRÜNTÜ BİRLEŞİMİYLE KARAKTER VE NESNE TANINMASI

ÖZ

Bu tezde ses ve görüntü tanıma yöntemlerini birleştiren yeni bir yaklaşım sunulmuştur. Insan beyni analiz edilmesi çok zor ve çok karmaşık bir yapıdadır. Fakat beynin bazı en temel öğrenme işlevleri simule edilebilir. Beynin temel öğrenme işlevleri içinde ses yada görüntü bilgisinin olmaması düşünülemez. Hem ses hem görüntü bilgisi beynin temel öğrenme işlevleri sırasında büyük rol oynamaktadır. Bu yüzden bu tez çalışmasında insan beyninin çalışmasına daha da yaklaşabilmek için ses ve görüntüyü entegre eden yeni algoritmalar denenmiştir. Algoritma ses ve görüntü bilgilerini ard arda alarak bu bilgilerden anlamlı tanımlama özellikleri çıkarıp bunları veri alanında yeni bir bilgi olarak entegre etmektedir. Yeni bilgi kaydedildikten sonra aynı bilgiyi temsil eden ses ve görüntü bilgileri algoritmayı eğitmek amacıyla kullanılmaktadır. Algoritmada kullanıcı eğitmen olarak rol almaktadır. Deneysel sonuçlar, öğrenme işlevlerinde algoritmanın uygulanabilir olduğunu göstermektedir.

Anahtar sözcükler : Ses tanıma, görüntü tanıma, ses ve görüntü birleşimi, yapay zeka.

CONTENTS

Page

CHAPTER ONE
INTRODUCTION

The learning tasks are very important tasks in the artificial intelligence applications. In every application the new methods are developed in order to make the machines learn more effectively. The one of the most difficult task is the speech recognition. The speech signal is very dynamic signal, and it changes according to the speaker, also speech signal shows meaningful differences when the same speaker speaks the same things. Therefore analyzing the speech signal is the most important and difficult task. But human brain performs these tasks in a very complicated way and in a very short time. The possibility of the brain to make an error is almost zero. Understanding some of the operations in the brain can be helpful in developing the new algorithms for artificial intelligence. The new and challenging approaches are introduced in the following paragraphs.

It is clear that brain operation is too complex, and simulating the brain is almost impossible. On the other hand there is no unique truth for human beings; the truth differs between the peoples. Because the absence of unique truth, the artificial intelligence algorithms cannot behave like a human brain, because the algorithms are developed by the human beings, the truths of the algorithm will be according to its developer. As a result the computers or other machines will not have their own free desires (Penrose, 1989).

On the other hand, the artificial intelligence defenders try to simulate the brain in a basic manner. Trying to make the exactly human like brain may be the impossible task; therefore the basic operations in the brain such as speech recognition, image recognition can be analyzed either in physiological or philosophical ways, and can be simulated in order to develop more efficient algorithms. Recent researches show that the audio-visual fusion is effective in increasing the system performance. The speech recognition systems provided with the speaker lip motion information are more effective systems (Ben-Yacoub, Abdeljaoued, & Mayoraz, 1999). For these purposes, some approaches to the operation of the human brain will be discussed in the following sections.

1.1 Human Brain

The information in the human brain is processed by the neurons. There are huge amount of neurons in the brain. The neurons consist of axon, synapse, and dendrite. The physiological structure of the brain and neurons are not going to be discussed, but some philosophical approaches will be explained. The artificial neural networks are used to simulate the neuron activity. The neurons are active or inactive according to the input information. If the weighted sums of the inputs exceed the threshold, then neurons are active and at their output the electrical pulses are generated. The artificial neural networks use the same idea as the real neurons. But there is an important difference between artificial and real neurons. The real neurons send information with the frequency. In other words, when the input signal is high the output frequency of the pulses is high (Crick, 1990). But the artificial neural networks does not produce the frequency output, the artificial neuron output is either 1 or 0 in discrete perceptron model. In adaptive linear elements, the output of the neuron is linear output of the weighted sum of the inputs. First the real neuron model is designed in this thesis work. The difference between the immediately following values of the signal is applied to the artificial neuron, and then the neuron produces the output 1 or 0 between the intervals specified by the input signal difference. When the input signal difference is high, the intervals of the 1's at the neuron output are small and vice versa. Then the output neurons are summed to simulate the other neuron inputs in the brain. The neurons in the brain are connected to each other in a very complicated way. After the summation, the new neuron output is produced according to the result of the summation in the preceding neuron. Then this neuron outputs are tried to be used as a features for the input signal. But unfortunately the system was not successful and needs more researches to be made, and also the operation time was high. But, if the well features from the neuron activity can be extracted, the system can be the very useful tool for image and speech recognition systems.

1.1.1 Human Brain Perception of Sound

The sound signal is very dynamic signal. The human brain perception of the sound is almost perfect. How the human brain performs these difficult tasks perfectly? If the child is considered, when he or she starts to learn, collects the sound input coming from the environment. Maybe the human speech sound is formed according to the speech sounds coming from the environment at the beginning of the learning process. For example, the child collects information from his/her father, mother, and so on. And his/her unique speech sound is produced according to the father or mother speech sound. The people living in different countries speak with very different speech sound. As an example, the people living in Japan can be recognized from his/her speech sound. Therefore the speech sound can be the average of the sound inputs to the human brain at the beginning of the learning process.

Another amazing property of the brain is to understand the speaker independent speech signals. For example, the letter ‘a’ shows difference in signal shape and frequency, but the brain understands the letter ‘a’ perfectly, no matter whether it is said by the child or old person. In this thesis, the discussed property of the brain is tried to be simulated. The all inputs coming from different speakers are collected, and then the selected features are updated according to the specified learning rules. The used learning rules are discussed in section 1.2.

In the perception of the human brain of the speech signal, the human ear input is divided into frequency bands (Figure 1.1) (Painter & Spanias, 2000). The signals at lower frequencies are divided into smaller bands, and the high frequencies to the wide bands. Every frequency band responds to the specified frequencies. These filter bands show that the information is more at lower frequencies. Because the brain uses these filter banks, the Fourier transform is used in this thesis work. The features are extracted from the Fast Fourier transform coefficients. The more features are extracted from the lower frequency bands. The means and variances of the specified fast Fourier transform band coefficients are used as a features for learning process.

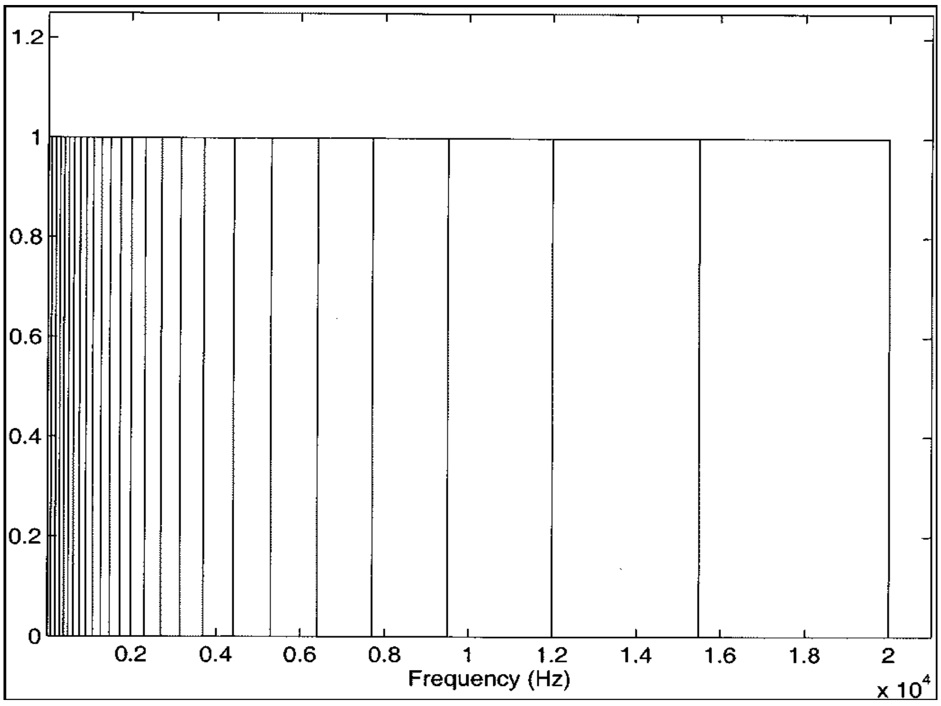


Figure 1.1 Idealized critical band filter bank

1.1.2 Human Brain Perception of Image

Another important property of the human brain is its ability to understand the images without an error. In the perception of the brain of the image data, the all image inputs are compared to the existing images in the brain, and then brain decides which one is the best matched image. The interesting example is given in figure 1.2 (Crick,1990).

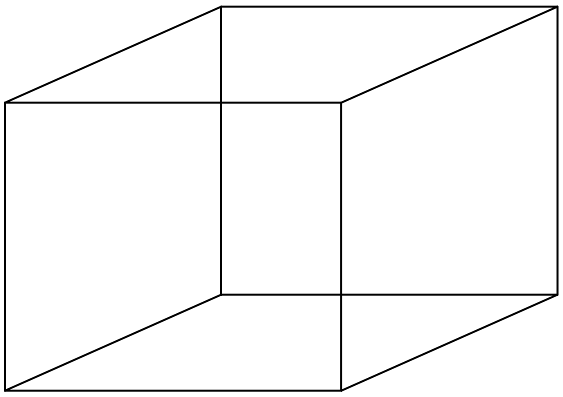


Figure 1.2 Shape changing cube

When someone looks carefully at this cube, the shape of the cube changes between the same time intervals. It looks left sided and downward cube for a small time interval, and for other small time interval it looks right sided and upward cube. This is because the brain knows these two shapes, and the best matches to this shape are same, and brain does not know which one is the best one, and therefore makes the two choices between the small time intervals. This example shows that there is a map in the human brain for the viewed images. In the image inputs from the human eye, the brain searches the existing map, and then decides the best matched shape (Crick, 1990).

The other important property of the human brain is its ability to find the edges of the objects. The edge detection is important for image processing applications. Some new approaches are going to be made for the human brain perception of images.

The human brain first detects the objects from the motion at the beginning of the learning processes. This approach is going to be supported with some examples. For example, the knife consists of wood and steel part. When the knife image input is present to the human brain, the brain understands it as a one object, but there is two parts of the same object and there exists one more edge between them. But the two parts of the knife are moving together; therefore the brain understands it as a one object. Another example can be the car. The car moves together with its parts, and human brain perception of the car is one object. The other parts of the car such as wheels can be removed from the car, and then because the wheels move independent from the car, the brain understands it as another object from the car. The most important properties of the objects are extracted from their motion by human brain. After the extractions are made, the brain can detect these objects no matter whether they move or not. This is because the human eye makes very small vibrations, and then all edges are extracted from the environment C. Guzelis (lesson discussion, March 2005), and then brain finds the best matches to the detected edges. This approach is tried to be simulated in this thesis work, and important results are obtained. The results of the experiments will be discussed in the chapter five. The following section will provide important approaches to the awareness in the human brain.

1.1.3 Awareness in the Human Brain

Another important property of the human brain is awareness. As discussed in the preceding sections, the human brain understands the speech signal and image data perfectly. But how the brain is aware of the viewed objects or heard sounds? In order to answer this question, the image and sound fusion must be considered first. The awareness cannot be explained without image or sound. For example, the pencil can be considered. When the image input of the pencil is present to the brain, then brain detects the edges, and then decides to the best match for the pencil. The brain is aware of the pencil by other information's provided with other sensory inputs, such as speech information. After the detection of the pencil, the sound information for the pencil is appeared inside the brain. Then brain decides whether to speak it or not. Also the other receptor inputs are important in order for the brain to be aware of the viewed objects or heard sounds. In the pencil example, the other information's can be the toughness of the pencil when touched by the fingers, or the information about what can be done by using the pencil. All these information's corresponds to one knowledge. When one of the sensory inputs, such as image or sound inputs, are present to the brain, then the other knowledges corresponding to the same knowledge are appeared in the brain. By collecting all these information's brain can speak the viewed object name, imagine the shape of the heard sound, or do something that can be done by the viewed object.

On the other hand, another interesting property of the human brain is to guess the exact image from one point of view. For example, human brain can recognize someone from his/her back view. Brain makes this guess by using the previously received information's for that person. Because brain collects the image inputs coming from the same person, and then saves them as a same knowledge in the memory. At this time the new question arises: How does brain decide that the received information's correspond to the same knowledge? Brain can decide it in a two way. First, if the object or person turns after the first view was saved, then brain can detect from motion that all image inputs taken at that time corresponds to the same knowledge. Second, if the first image input from the object or person is taken only from one point of view, and at another time the different view of the object or person is taken as image input, then brain can decide that this image input is same knowledge as before viewed object or person by using the speech information. Because, the speech information is unique information corresponding to any knowledge. Therefore the role of the speech signals and image data in learning process are very important.

1.2 Kohonen's Self Organizing Map

The Kohonen's self organizing map is the artificial neural network algorithm used as an unsupervised learning process, that is the desired output isn't present to the system. The artificial neural network algorithms such as back propagation are supervised learning algorithms, because the desired outputs are given to the system, and then the algorithm learns from input sequence to produce the desired output by adjusting its weights.

The important uses of Kohonen's self organizing map are the vector quantization (Khaparde & Gandhi, 1993), and topological feature map (Ichiki, Hagiwara & Nakagawa, 1993). The Kohonen's network can be used in classification problems.

In this thesis work the modified Kohonen's Self Organizing Map algorithm is used as a supervised learning algorithm. The algorithm will be discussed in detail in chapter two. By using the modified Kohonen's Self Organizing Map algorithm, the fusion of image and sound is realized, and the system is trained by the user in order to improve the efficiency of speech and image recognition tasks.

CHAPTER TWO
UNSUPERVISED LEARNING IN NEURAL NETWORKS

Unsupervised learning is considered to be more psychologically plausible, possibly because humans tend to learn more about nature and life through their own experience, rather than by listening to a teacher. For example, imagine that there is nobody to tell us which instance to classify into which class, that is, what we should derive. The patterns in this case are not labeled with class-labels (Kasabov, 1996).

2.1 Unsupervised Learning Algorithms

Two principle learning rules are implemented in the contemporary unsupervised learning algorithms for neural networks: (1) noncompetitive learning, that is, many neurons may be activated at a time; and (2) competitive learning, that is, the neurons compete and after that, only one is activated at one time, e.g. only one wins after the competition. This principle is also called "winner takes all" (Kasabov, 1996).

Most of today's training algorithms are influenced by the concept introduced by Donald O. Hebb (1949). Hebb proposed a model for unsupervised learning in which the synaptic strength (weight) is increased if both the source and destination neurons are simultaneously activated. It is expressed as:

$$w_{ij}(t+1) = w_{ij}(t) + c \cdot o_i \cdot o_j \cdot \tag{2.1}$$

where $w_{ij}(t)$ is the weight of the connection between the ith and jth neuron at the moment t, and o_i and o_j are the output signals of neurons i and j at the same moment t. The weight $w_{ij}(t + 1)$ is the adjusted weight at the next moment (t + 1). This principle was used in supervised learning, but then we knew the desired outputs. Here, the outputs are as they are produced by the network only. Different modifications of this rule have been suggested, for example: the differential Hebbian learning law (Kosko, 1988):

$$w_{ij}(t+1) = w_{ij}(t) + c \cdot o_i \cdot o_j + \Delta o_i \cdot \Delta o_j \tag{2.2}$$

The differential Hebbian law introduces the first derivatives of the activation signals to the Hebbian law.

- Grossberg's competitive law (Grossberg 1982), expressed as:

$$\Delta w_{ij} = c \cdot o_j (o_i \cdot w_{ij}) \tag{2.3}$$

- The differential competitive learning law (Kosko 1990):

$$\Delta w_{ij} = c \cdot \Delta o_j \cdot (o_i - w_{ij}) \tag{2.4}$$

The differential competitive learning law introduces the first derivative of neuronal output values to the competitive learning law.

- Adaptive vector quantization (Kohonen 1982, 1990); the learning law in a vector form is:

$$w_j(t+1) = w_j(t) + c \cdot (x(t) - w_j(t)) \tag{2.5}$$

where c is a learning rate, x(t) is the input vector at moment t, and w_j is the vector of the weights from the input neurons to the neuron j.

Unsupervised learning is also applicable when noise is present. For example, the random signal Hebbian law looks like (Kasabov, 1996):

$$w_{ij}(t+1) = w_{ij} + c \cdot o_i \cdot o_j + n_{ij} \tag{2.6}$$

where n_{ij} is a noise introduced to the connection i-j.

In the major competitive learning models, the neural networks are organized in layers. There are excitatory connections between neurons from different layers, and inhibitory connections between neurons in one layer. The neurons in the output layer compete, each one inhibiting the others with its current activation level. The winner takes all in the end, that is, the neuron that gets the highest activation level is activated. Then a "reward" follows; this neuron strengthens its weights according to the input pattern and possibly suppresses the other neurons' weights.

Unsupervised learning is applicable for conceptualization that is, discovering and creating new concepts and categories from data examples. Unsupervised learning in neural networks can be used to learn structures, similarities, and relations.

Another major application for unsupervised learning networks is vector quantization. N-dimensional data vectors are represented as k-dimensional, when k < n. This is important for signal compression (images, speech, etc.), clustering, reducing dimensionality, etc (Kasabov, 1996).

2.2 Kohonen Self Organizing Topological Map

One of the most used neural network models, used mainly for vector quantization and data analysis but also applicable to almost all the tasks where neural networks have been tried successfully, is the self organizing map introduced and developed by Teuvo Kohonen (1982, 1990, 1993).

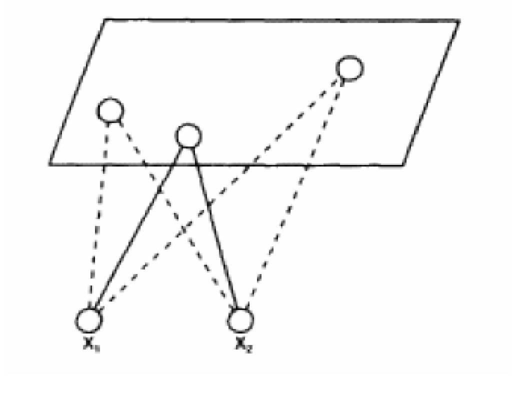


Figure 2.1 A part of a Kohonen SOM

2.2.1 The Philosophy of the Kohonen Network

Self-organizing topological or feature maps became popular because they tend to be very powerful at solving problems based on vector quantization. A SOM consists of two layers, an input layer and an output layer, called a feature map, which represent the output vectors of the output space (figure 2.1). The weights of the connections of an output neuron j to all the n input neurons form a vector w_j in an n dimensional space. The input values may be continuous or discrete, but the output values are binary only. The main ideas of SOM are as follows (Kasabov, 1996):

- Output neurons specialize during the training or recall procedure (both may not be distinguished here) to react to input vectors of some groups (clusters) and to represent typical features shared by the input vectors. This characteristic of the SOM tends to be biologically plausible as there is evidence to show that the brain is organized into regions that correspond to different sensory stimuli. There is also evidence for linguistic units being locatable within the human brain. A SOM is able to extract abstract information from multidimensional primary signals and to represent it as a location, in one-, two-, three-, etc. dimensional space (Kohonen 1990).

- The neurons in the output layer are competitive. Lateral interaction between neighboring neurons is introduced in such a way that a neuron has a strong excitatory connection to itself and fewer excitatory connections to its neighboring neurons within a certain radius; beyond this area, a neuron either inhibits the activation of the other neurons by inhibitory connections, or does not influence it. One possible rule is the so-called Mexican hat rule. In general, this is "the winner-takes-all" scheme, where only one neuron is the winner after an input vector has been fed in and a competition between the output neurons has taken place. The winning neuron represents the class, the label, and the feature to which the input vector belongs.

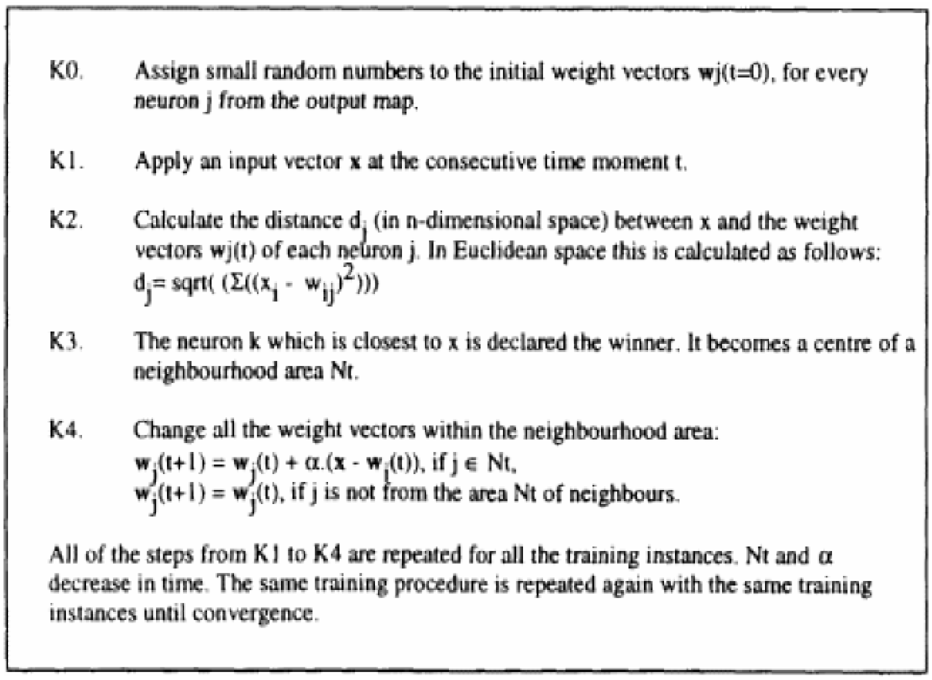
- The SOM transforms or preserves the similarity between input vectors from the input space into topological closeness of neurons in the output space represented as a topological map. Similar input vectors are represented by near points (neurons) in the output space. The distance between the neurons in the output layer matters, as this is a significant property of the network.

There are two possibilities for using the SOM. The first is to use it for the unsupervised mode only, where the labels and classes that the input vectors belong to are unknown. A second option is when SOM is used for unsupervised training followed by a supervised training. LVQ (learning vector quantization) algorithms (Kohonen 1990) have been developed for this purpose. In the LVQ algorithms, after an initial unsupervised learning is performed in a SOM, the output neurons are calibrated, labeled for the known classes, and a supervised learning is performed which refines the map according to what is known about the output classes, regions, groups, and labels. The two possibilities are presented and discussed in the next subsections (Kasabov, 1996).

2.2.2 Unsupervised Self – Organizing Feature Map

The unsupervised algorithm for training a SOM, proposed by Teuvo Kohonen, is outlined in figure 2.2. After each input pattern is presented, the winner is found and the connection weights in its neighborhood area Nt increase, while the connection weights outside the area are kept as they are. α is a learning parameter. It is recommended that the training time moments t (cycles) are more than 500 times the output neurons. If the training set contains fewer instances than this number, then the whole training set is fed again and again into the network.

SOM’s learn statistical features. The synaptic weight vectors tend to approximate the density function of the input vectors in an orderly fashion (Kohonen 1990). Synaptic vectors wj converge exponentially to centers of groups of patterns and the whole map represents to a certain degree the probability distribution of the input data. This property of the SOM is illustrated by a simple example of a two input, 20



K0. Assign small random numbers to the initial weight vectors wj(t=0), for every neuron j from the output map.

K1. Apply an input vector **x** at the consecutive time moment t.

K2. Calculate the distance d_j (in n-dimensional space) between x and the weight vectors wj(t) of each neuron j. In Euclidean space this is calculated as follows:
$d_j= \mathrm{sqrt}((\Sigma((x_i - w_{ij})^2)))$

K3. The neuron k which is closest to x is declared the winner. It becomes a centre of a neighbourhood area Nt.

K4. Change all the weight vectors within the neighbourhood area:
$\mathbf{w}_j(t+1) = \mathbf{w}_j(t) + \alpha.(\mathbf{x} - \mathbf{w}_j(t))$, if $j \in$ Nt,
$\mathbf{w}_j(t+1) = \mathbf{w}_j(t)$, if j is not from the area Nt of neighbours.

All of the steps from K1 to K4 are repeated for all the training instances. Nt and α decrease in time. The same training procedure is repeated again with the same training instances until convergence.

Figure 2.2 A realization of the Kohonen SOM algorithm

x 20 -output SOM. The input vectors are generated randomly with a uniform probability distribution function as points in a two-dimensional plane, having values in the interval [0, 1]. Figure 2.3 represents graphically the change in weights after some learning cycles. The first box is a two dimensional representation of the data. The other boxes represent SOM weights also in a two dimensional space. The lines connect neighboring neurons, and the position of a neuron in the two dimensional graph represents the value of its weight vector. Gradually the output neurons become self organized and represent a uniform distribution of input vectors in the input space. The time (in cycles) is given in each box as well. If the input data have a well-defined probability density function, then the weight vectors of the output neurons try to imitate it, regardless of how complex it is. The weight vectors are also called reference vectors or reference codebook vectors, and the whole weight vector space is called a reference codebook. Figure 2.4 shows the adaptation of the same network to the generated input vectors in 2 when the latter are not uniformly distributed but have a Gaussian distribution of a star type (Kasabov, 1996).

The SOM is very similar to elastic net which covers the input vector's space.

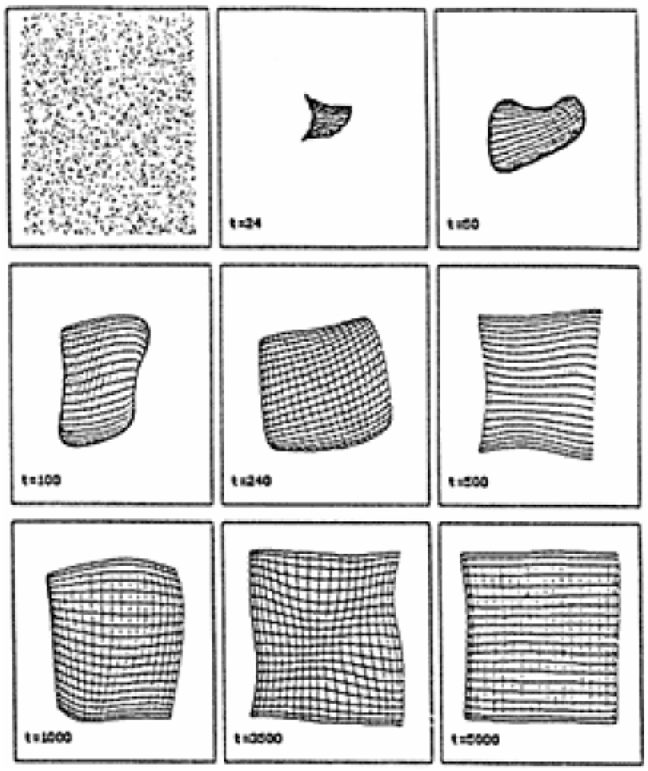


Figure 2.3 Uniformly randomly distributed two-dimensional vectors are learned by a two-dimensional SOM after 5000 iterations.

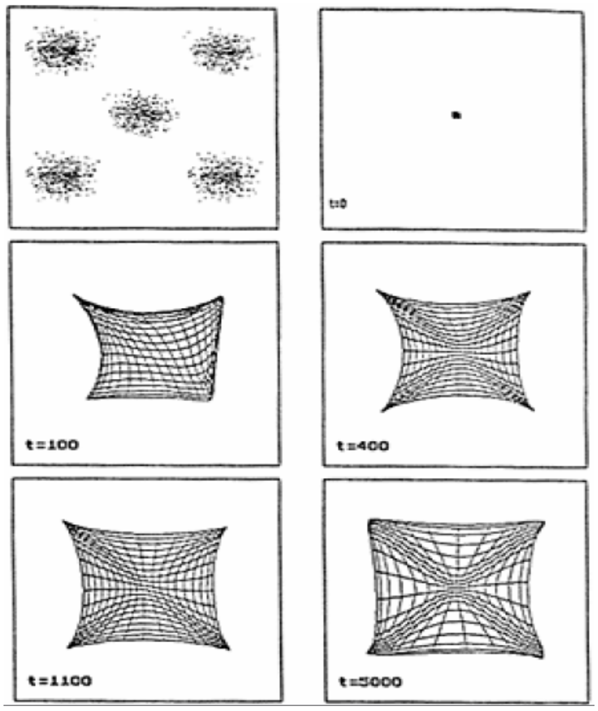


Figure 2.4 Learning Gaussian distributed random vectors by a two-dimensional SOM.

2.2.3 Learning Vector Quantization Algorithms for Supervised Learning

The problem of distinguishing input vectors that "fall" in the bordering area between two output neurons and the necessity for using known labels of the neurons for better topological mapping of the input vectors into the output space have led to the development of learning vector quantization (LVQ) algorithms LVQ1, LVQ2, and LVQ3 (Kohonen 1990).

- *LVQ1 Algorithm:* In LVQ1 several codebook vectors are assigned to each class, and each is labeled with the corresponding class symbol (label). Initial values of the codebook vectors are learned by the SOM algorithm. The output neurons are then labeled according to the classes. Then a correction of the weights with the use of the known labels is performed by applying the following formulas to update the weights vectors:

$$w_j(t+1) = w_j(t) + \alpha \cdot (x(t) - w_j(t)) \tag{2.7}$$

if x is classified by the network correctly in class c_j represented by the jth neuron

$$w_j(t+1) = w_j(t) - \alpha \cdot (x(t) - w_j(t)) \tag{2.8}$$

if x has been wrongly associated by the network with class-neuron j

$$w_j(t+1) = w_j(t) \tag{2.9}$$

for all i different form j. $\alpha(t)$ is a scalar gain factor which decreases monotonically in time.

- *LVQ2 Algorithm:* In the LVQ2 algorithm, a further tuning of the weights of the immediate neighbors to the winning neuron, after a presentation of an input vector, is done. If, for example, c_i is the class of the winning neuron, but x belongs to the class cj of its neighbor j, the following formulas are applied to calculate the new weight vectors for neurons i and j:

$$w_i(t+1) = w_i(t) - \alpha \cdot (x(t) - w_i(t)), \text{ or} \quad (2.10)$$

$$w_j(t+1) = w_j(t) + \alpha \cdot (x(t) - w_j(t)), \text{ or} \quad (2.11)$$

$$w_k(t+1) = w_k(t) \quad (2.12)$$

for the rest of the neurons k.

- *LVQ3 Algorithm:* In the LVQ3 algorithm a window between the lateral output neurons i and j is considered. When the input vector x falls out of the window, it is the LVQ2 algorithm which determines one of the i or j neurons to be a winner and updates the new codebook vectors. But if the input vector x falls in the window, then other formulas are applied:

$$w_i(t+1) = w_i(t) - \alpha \cdot (x(t) - w_i(t)) \quad (2.13)$$

if x falls in the "window" and x belongs to class c_j

$$w_j(t+1) = w_j(t) + \alpha \cdot (x(t) - w_j(t)) \quad (2.14)$$

for k ∈ {i, j}, if x falls in the "window" and x belongs to class c_j,

$$w_k(t+1) = w_k(t) + \alpha \cdot (x(t) - w_k(t)) \quad (2.15)$$

for k ∈ {i, j}, if x falls in the window and i and j represent the same class.

2.3 A Novel Approach: The Supervised Organizing Map Algorithm

As discussed in the preceding sections, the unsupervised learning algorithm can be adopted as an input to the supervised learning algorithm. The Kohonen feature maps can be used to extract the features from the input vector, and then fed to the supervised learning algorithm such as Back Propagation (Ichiki, Hagiwara & Nakagawa, 1993).

In this thesis work the modified Kohonen Self Organizing Map algorithm is applied to the input vectors, which is different from the above discussed algorithms. The modified Kohonen Self Organizing Map algorithm is used as a supervised learning model for sound and image input vectors. In “winner takes all” algorithm the winning neuron is given chance to update its weight vector, but in the modified algorithm the winning neuron is defined by the teacher (user). First, the algorithm finds the minimum distance from input vector to weight vector and then defines the winning neuron, but before updating the weights of the winning neuron, asks to the user whether it is correct class or not. If the user confirms the winning neuron then the weights of the winning neuron are updated, but if the user doesn’t confirm the winning neuron, then the second winning neuron is given chance to update its weights after the user confirmation. If the second winning neuron is not correct class, then algorithm continues until the correct neuron will win. But if input vector doesn’t belong to any class defined by the weights of the neurons, then that input is used as a new knowledge to the algorithm, and new class is defined by using the input vector as a weight vector for that class. After the several execution of the algorithm the classes defined by the weight vector of the neurons are classified according to the information received from the teacher. The weight vector update equations are as follows:

distance: $d = \left| x(t) - w_j(t) \right|$ (2.16)

$k = \min(d)$ (2.17)

$$w_k(t+1) = w_k(t) + \alpha \cdot (x(t) - w_k(t)) \qquad (2.18)$$

if input vector x is classified correctly by the c_k neuron (i.e. if the user confirms the winning neuron),

$$d(k) = 10^{15} \qquad (2.19)$$

$$k = \min(d) \qquad (2.20)$$

$$w_k(t+1) = w_k(t) + \alpha \cdot (x(t) - w_k(t)) \qquad (2.21)$$

if input vector x is classified correctly by the c_k neuron (i.e. if the user confirms the second winning neuron).

The algorithm continues to find the winning neuron by adjusting the distance‘d’ to maximum for the winning neuron that is not confirmed by the user. If the user does not confirm any of the winning neurons, the new class weight vector is assigned to the network by adjusting initial weights to the input vector.

CHAPTER THREE
SOUND AND IMAGE PROCESSING TECHNIQUES

3.1 Introduction to Speech Processing Tasks

Speech processing includes different technologies and applications. Some of them, according to Morgan and Scofield (1991), are listed below:

- Speech encoding aims at voice transmission, speech compression, and secure communications.

- Speaker separation aims at extracting speech signals of each of the speakers when multiple talkers are present.

- Speech enhancement aims at improving the intelligibility of the speech signals.

- Speaker identification aims at "identifying an uncooperative talker in an environment where a large number of talkers may be present."

- Language identification aims at discriminating languages.

- Keyword spotting, that is, recognizing spoken keywords from a dictionary (for database retrieval, etc).

But the most interesting and most rapidly developing of the speech-processing problems is the automatic speech recognition (ASR) problem. It aims at providing enhanced access to machines via voice commands. A voice interface to a computer is related strongly to analysis of the spoken language, concept understanding, intelligent communication systems, and further on, to developing "consciousness" in the machines. These are challenging problems for the AI community.

The elaboration of practical systems for speech recognition takes two major trends: (1) recognition of separately pronounced words in extended speech; (2) recognition and comprehension of continuous speech.

Two approaches are mainly used in ASR: global and analytical. The global approach is based on comparison of the whole word with standard patterns, whereas in the analytical approach a word is broken into segments (sub words, units) on the basis of the phonetic characteristics of the speech signal. In both global and analytical approaches, obtained parametric vectors from the speech signal must be classified. A parametric vector of n elements can be represented as a point in n-dimensional space. This point can be seen as a pattern (Kasabov, 1996).

3.1.1 The Nature of Speech

Speech is a sequence of waves which are transmitted over time through a medium and are characterized by some features, including intensity and frequency. Speech is perceived by the inner ear in humans. It activates oscillations of small elements in the media of the inner ear, which oscillations are transmitted to a specific part of the brain for further processing. The biological background of speech recognition is used by many researchers to develop humanlike ASR systems, but other researchers take other approaches. Speech can be represented on the (Kasabov, 1996):

- Time scale, which representation is called a waveform representation
- Frequency scale, which representation is called a spectrum
- Both a time and frequency scale, which is the spectrogram of the speech signal.

3.1.2 Variability in Speech

The fundamental difficulty of speech recognition is that the speech signal is highly variable according to the speaker, speaking rate, context, and acoustic conditions.

There are a great number of factors which cause variability in speech such as the speaker, the context, and the environment. The speech signal is very dependent on the physical characteristics of the vocal tract, which in turn depend on age and gender. The country of the speaker and the region in the country the speaker is from can also affect speech. Different accents of English can mean different acoustic realizations of the same phonemes. If English is the second language of the speaker, there can be an even greater degree of variability in the speech.

The same speaker can show variability in his or her speech, depending on whether it is a formal or informal situation. People speak precisely in formal situations and imprecisely in informal situations because the speaker is more relaxed. The more familiar a speaker is with a computer speech recognition system, the more informal his or her speech becomes, and the more difficult for the speech recognition system to recognize the speech. This could pose problems for speech recognition systems if they could not continually adjust.

Words may be pronounced differently depending on their context. Words are pronounced differently depending on where they lie in a sentence and the degree of stress placed upon them. In addition, the speaking rate can cause variability in speech. The speed of speech varies according to such things as the situation and emotions of the speaker. The duration of sounds in fast speech, however, do not reduce proportionately to their duration in slow speech (Kasabov, 1996).

3.1.3 Factors That Influence the Performance of the ASR System

All the speech recognition tasks are constrained in order to be solved. Through placing constraints on the speech recognition system, the complexity of the speech recognition task can be considerably reduced. The complexity is basically affected by (Kasabov, 1996):

- *Vocabulary size* (the range of words and phrases the system understands). Many tasks can be performed with a small vocabulary, although ultimately the most useful systems will have a large vocabulary. In general, vocabulary size is as follows:

  Small, tens of words.
  Medium, hundreds of words.
  Large, thousands of words.
  Very large, tens of thousands of words.

- *The speaking format of the system*, that is,

  Isolated words (phrase) recognition.
  Connected word recognition; this uses fluent speech but a highly constrained vocabulary, for example, digit dialing.
  Continuous speech recognition.

- The degree of speaker dependence of the system, that is, whether it is:

  Speaker-dependent (trained to the speech patterns of an individual user).
  Multiple speakers (trained to the speech patterns of a limited group of people).
  Speaker-independent (such a system could work reliably with speakers who have never used the system).

- The constraints of the task, that is, as the vocabulary size increases, the possible combinations of words to be recognized grow exponentially. Some form of task constraint, such as formal syntax and formal semantics, is required to make the task more manageable.

3.1.4 Building ASR Systems

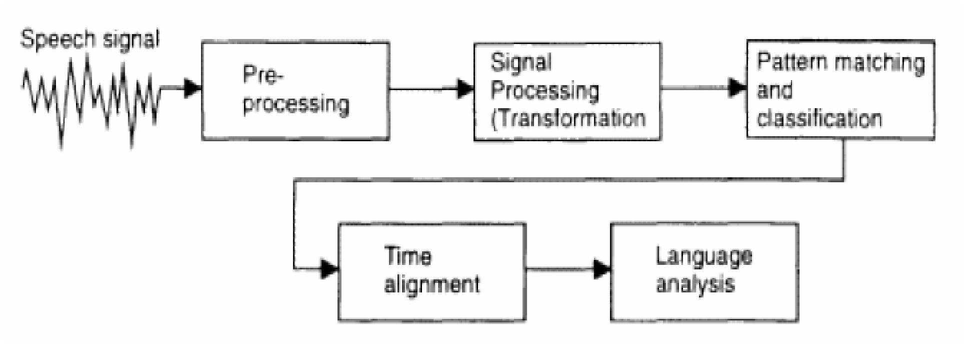

Speech signal
Pre-processing
Signal Processing (Transformation
Pattern matching and classification
Time alignment
Language analysis

Figure 3.1 Main blocks in speech recognition systems

Figure 3.1 shows a simplified diagram of a computer speech recognition system. It comprises five major blocks (Kasabov, 1996):

1. *Preprocessing*—sampling and digitizing the signal.

2. *Signal processing*—transforming the signal taken for a small portion of time into an n-dimensional feature vector, where n is the number of features used (fast Fourier transform, mel-scaled cepstrum coefficient).

3. *Pattern matching*—matching the feature vector to already existing ones and finding the best match.

4. *Time alignment*—a sequence of vectors recognized over a time are aligned to represent a meaningful linguistic unit (phoneme, word).

5. *Language analysis*—the recognized language units recognized over time are further combined and recognized from the point of view of the syntax, the semantics, and the concepts of the language used in the system.

3.1.5 Language Analysis. The Turing Test for AI

Natural language understanding is an extremely complex phenomenon. It involves recognition of sounds, words, and phrases, as well as their combining forms, comprehension, and usage. There are various levels in the process of language analysis (Kasabov, 1996):

- *Prosody* deals with rhythm and intonation.
- Phonetics deals with the main sound units of speech (phonemes) and their correct combination.
- Lexicology deals with the lexical content of language.
- Semantics deals with the meaning of words and phrases seen as a function of the meaning of their constituents.
- Morphology deals with the semantic components of words (morphemes).
- Syntax deals with the rules, which are applied for matching words in order to form sentences.
- Pragmatics deals with the modes of use of language and their impact on the listener.

The importance of language understanding in communication between humans and computers was the essence of the Turing test for AI. Alan Turing, a British mathematician and computer scientist, introduced a definition of AI. A machine system is considered to possesses AI if, while communicating with a person behind a "bar," the person cannot recognize whether it is a machine or a human. To put it more simply, an AI system is a system that can communicate with humans in their natural language. This test has not been passed by any computer system so far, and it is unlikely to be passed by any machine in the near future.

Computer systems for language understanding require methods which can represent ambiguity, commonsense knowledge, and hierarchical structures. Humans, when communicating with one another, share a lot of commonsense knowledge which is inherited and learned in a natural way. This is a problem for a computer program. Humans use facial expressions, body language, gestures, and eye movements when they communicate. So they communicate in a multimodal manner. Computer systems that analyze speech signals, gestures, and facial expressions when communicating with users are called multimodal interfaces (Kasabov, 1996).

3.1.6 Intelligent Human-Computer Interfaces

One application of speech recognition and language modeling systems is for building intelligent human computer interfaces (IHCI). A general block diagram of an IHCI is graphically depicted in figure 3.2. The system allows for retrieving information from a database or for connecting the user to other communication ports by using both speech and text. It consists of the following major modules (Kasabov, 1996):

- *Speech recognition and language modeling block*. This module is trained on a speech corpus and uses a linguistic knowledge corpus. It recognizes spoken words, phrases, and sentences from a defined dictionary. The input speech signals are first digitized, transformed into feature vectors, and then used in the speech recognition sub module to match with already known speech patterns and to recognize small speech units, for example, phonemes. The language modeling sub module takes the recognized speech units and combine them into meaningful and relevant words and sentences.

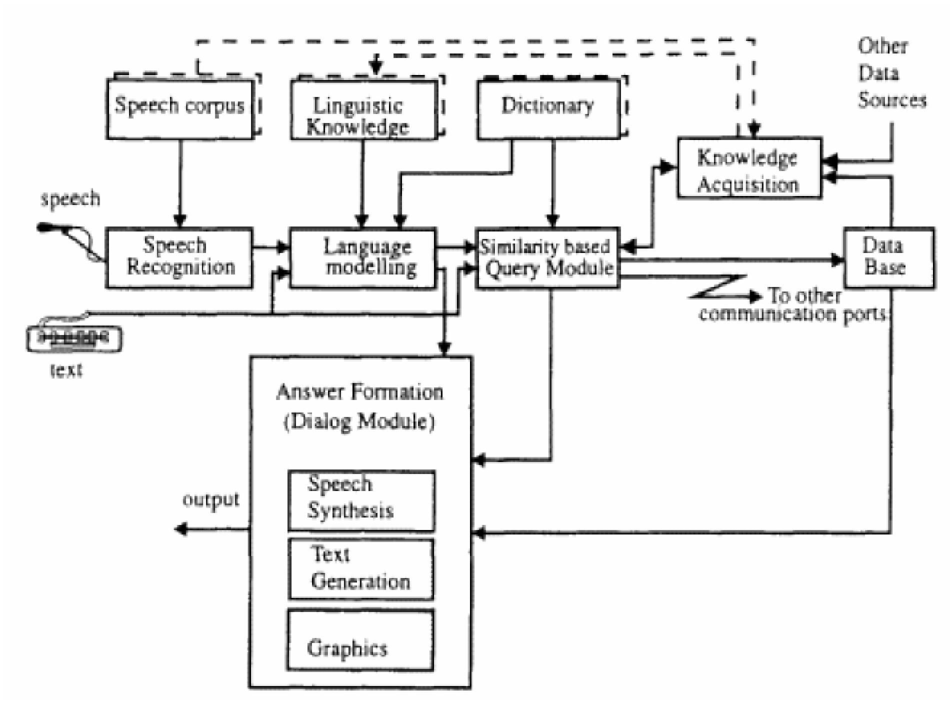


Figure 3.2 A block diagram of an intelligent human-computer interface (IHCI)

- *Similarity-based query module.* This module does approximate reasoning over user's query and allows for vague, fuzzy queries to be used. For example, the user can ask for a list of patients who have "high blood pressure" when the database contains the blood pressure parameter in a numerical form only. The module performs two levels of matching a query to a database. The first one is exact matching, when the query matches exactly the information in the database. The second one is similarity-based matching, which involves interaction between the user and the system. The module finds language or conceptual similarities based on previous knowledge represented in terms of fuzzy rules. Fuzzy logic can be used for implementing the module as fuzzy systems have proved to be good at representing humanlike reasoning based on similarities.

- *Knowledge acquisition module.* The module is used to extract knowledge from different sources of information, for example, for extracting linguistic rules from linguistic database or extracting trading rules from a stock exchange database.

- *Answer formation module.* This module produces the answer to the user and performs a dialogue at any phase of the information retrieval. It has speech synthesis and text generation sub modules.

3.2 Introduction to Digital Image Processing

Image processing is a subclass of signal processing concerned specifically with pictures. Improve image quality for human perception and/or computer interpretation (Gonzalez, 1992).

3.2.1 Digital Image Representation

The term image refers to a two dimensional light intensity function f(x,y); (x,y) denotes spatial coordinates and f(x,y) is proportional to the brightness (gray-level) of the image at that point.

A digital image is an image f(x,y) that has been discretized both in spatial coordinates and brightness. It can be considered a matrix whose row and column indices identify a point in the image, and the corresponding matrix element value identifies the gray level at that point. The elements of such digital array are called image elements or picture elements, or pixels (Gonzalez, 1992).

3.2.2 Fundamental Steps in Image Processing

- *Image Acquisition*: an image is captured by a sensor (such as a monochrome or color TV camera) and digitized. If the output of the camera or sensor is not already in digital form, an analog-to digital converter digitizes it.
- *Frame Grabber*: Frame grabber only needs circuits to digitize the electrical signal from the imaging sensor to store the image in the memory (RAM) of the computer.
- *Preprocessing*: to improve the image quality to increase the chances for success of the other processes.

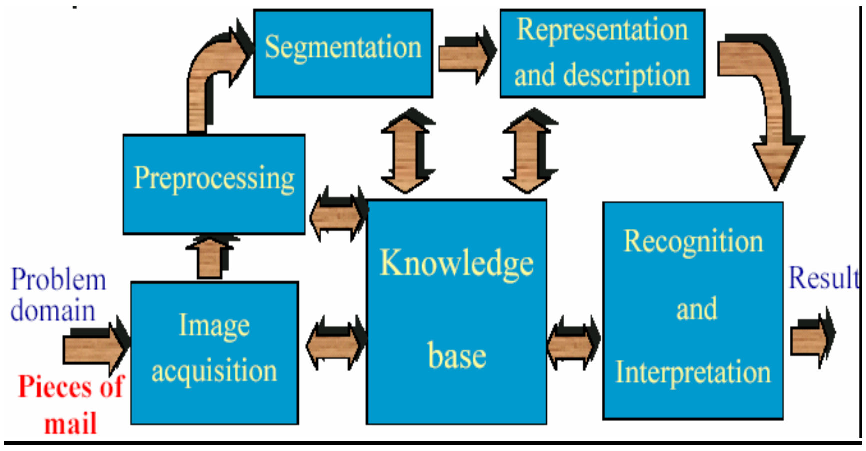


Figure 3.3 Example of digital image processing techniques for automatically reading the address on pieces of mail.

- *Segmentation*: Partitions an input image into its constituent parts or objects.
- *Representation*: transforms raw data (output of segmentation) into a form suitable for subsequent computer processing.
- *Description*: Deals with extracting features for differentiating one class of objects from another.
- *Recognition*: is the process that assigns a label to an object based on the information provided by its descriptors.
- *Interpretation*: involves assigning meaning to an ensemble of recognized objects.

3.2.3 A simple Image Model

Image refers to a 2D light-intensity function, f(x,y). The amplitude of f at spatial coordinates (x,y) gives the intensity (brightness) of the image at that point. Light is a form of energy thus f(x,y) must be nonzero and finite. $0 < f(x,y) < \infty$

The basic nature of f(x,y) may be characterized by 2 components (Gonzalez, 1992):

- The amount of source light incident on the scene being viewed: Illumination, i(x,y)
- The amount of light reflected by the objects in the scene: Reflectance, r(x,y)

$$f(x, y) = i(x, y)\, r(x, y) \tag{3.1}$$
$$0 < i(x, y) < \infty$$
$$0 < r(x. y) < 1$$

3.2.4 *Some Basic Relationship Between Pixels*

- *Neighbors of a pixel:* A pixel at coordinates (x,y) has four horizontal and vertical neighbors whose coordinates are given by (x+1, y), (x-1, y),(x, y+1), (x, y-1). This set is called 4-neighbors of p and denoted by $N_4(p)$. The four diagonal neighbors of p have coordinates (x+1, y+1), (x+1, y-1), (x-1, y+1), (x-1, y-1), and are denoted by $N_d(p)$. These points together with the 4-neighbors are called the 8-neighbors of p, denoted by $N_8(p)$.

- *Connectivity:*: Let V be the set of gray-level values used to define connectivity.
- *4-connectivity:* 2 pixels p and q with values from V are 4-connected if q is in the set $N_4(p)$.
- *8-connectivity:* 2 pixels p and q with values from V are 8-connected if q is in the set $N_8(p)$.
- *m-connectivity (mixed connectivity):* 2 pixels p and q with values from V are m-connected if q is in the set $N_4(p)$ or q is in the set $N_d(p)$ and the set $N_4(p) \cap N_8(q)$ is empty (the set of pixels that are 4-neighbors of both p and q whose values are from V).

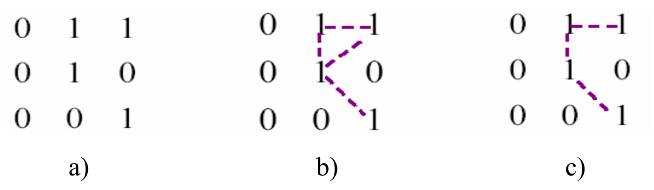


Figure 3.4 a) Arrangement of pixels, b) 8-neighbors of the center pixel, c) m-neighbors of the center pixel.

m-connectivity eliminates the multiple path connections that arise in 8-connectivity.

- *Adjacent:* a pixel p is adjacent to a pixel q if they are connected. Two image area subsets S1 and S2 are adjacent if some pixel in S1 is adjacent to some pixel S2.

- *Path:* a path from pixel p with coordinates (x,y) to pixel q with coordinates (s,t) is a sequence of distinct pixels with coordinates (x0, y0), (x1, y1),…(xn,yn) where (x0, y0) = (x, y) , (xn, yn) = (s, t) and (xi, yi) is adjacent to (xi-1, yi-1). n is the length of the path. We can define 4-,8-, or m-paths depending on type of adjacency specified.

- *Distance Measures:* Pixels p with the coordinates (x,y) and q with the coordinates (s,t).
- *Euclidean distance between p and q:*

$$De\,(p,q) = [\,(x - s)^2 + (y - t)^2\,]^{1/2} \qquad (3.2)$$

p : central pixel, q : any neighbor of p.

De (p, q) around p:

$$\begin{array}{ccc} 2^{1/2} & 1 & 2^{1/2} \\ 1 & 0 & 1 \\ 2^{1/2} & 1 & 2^{1/2} \end{array}$$

- *City-block distance between p and q:*

$$D_4\,(p, q) = |\,x - s\,| + |\,y - t\,| \qquad (3.3)$$

D_4 (p, q) around p:

$$\begin{array}{ccc} 2 & 1 & 2 \\ 1 & 0 & 1 \\ 2 & 1 & 2 \end{array}$$

- *Chessboard distance between p and q:*

$$D_8(p, q) = \max(|x - s|, |y - t|) \tag{3.4}$$

$D_8(p, q)$ around p:

$$\begin{matrix} 2 & 2 & 2 & 2 & 2 \\ 2 & 1 & 1 & 1 & 2 \\ 2 & 1 & 0 & 1 & 2 \\ 2 & 1 & 1 & 1 & 2 \\ 2 & 2 & 2 & 2 & 2 \end{matrix}$$

- *Arithmetic / Logic Operations:* are used extensively in most branches of image processing, and carried out pixel by pixel.

  Addition: p + q – The principal use is for image averaging to reduce noise.

  Subtraction: p – q – To remove static background info. (example: in medical imaging).

  Multiplication: p x q – To correct gray level shading resulting from non-uniformities in illumination or in the sensor.

  Division: p / q – To correct gray level shading resulting from non-uniformities in illumination or in the sensor.

- *Logic Operations:*

AND	: p AND q	(p x q)
OR	: p OR q	(p + q)
COMPLEMENT	: NOT q	(q)`
XOR	: p XOR q	(p + q)

3.2.5 Neighborhood Oriented Operations

Arithmetic and logic operations also are used in neighborhood oriented operations. Neighborhood processing is performed by so called mask operations. The terms template, window, and filter also are used to denote a mask (Gonzalez, 1992).

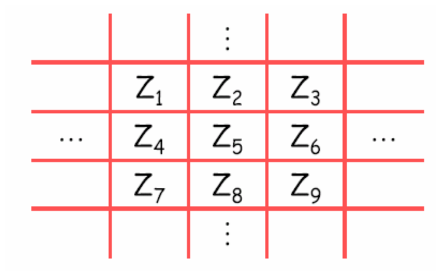


Figure 3.5 A 3 x 3 image region

Consider the sub image area shown above and suppose that we want to replace the value of Z_5 with the average value of the pixels in a 3 x 3 region centered at the pixel with value. To do so the arithmetic operation is in the form

$$Z = 1/9\ (Z_1 + Z_2 + \ldots\ldots + Z_9) \qquad (3.5)$$

and assign to Z_5 the value of Z. The same operation can be obtained in more general terms by centering a mask, shown below, at Z_5 multiplying each pixel under the mask by corresponding coefficient, and adding the results.

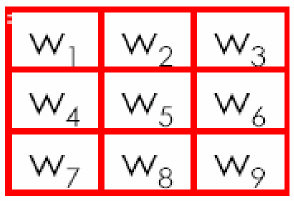


Figure 3.6 3 x 3 mask

$$Z = (w_1Z_1 + w_2Z_2 + \ldots\ldots + w_9Z_9) \qquad (3.6)$$

If we let $w_i = 1/9$, this operation gives the same result as the averaging procedure just discussed. A variety of useful image operations are applied by proper selection of mask coefficients, such as noise reduction, edge detection, image sharpening. However, applying a mask at each pixel location in an image is a computationally expensive task. Most modern image processors are equipped with an ALU, whose function is to perform arithmetic and logic operations in parallel typically at video frame rates (1/25s) (Gonzalez, 1992).

3.2.6 Image Enhancement

The principal objective of enhancement is to process an image so that the result is more suitable than the original image for a specific application.

The approaches fall into 2 categories (Gonzalez, 1992):

1) Spatial domain methods (refers to image plane itself, direct manipulation of pixels in an image).
2) Frequency domain methods (based on modifying the FT of an image).

Image processing functions in the spatial domain may be expressed as

$$g(x,y) = T[f(x,y)] \tag{3.7}$$

where f(x, y) is the input image, g(x, y) is the processed image, and T is an operator on f, defined over some neighborhood of (x, y). In addition, T can also operate on a set of input images, such as performing the pixel-by-pixel sum of M images for noise reduction. A square or rectangular sub image area centered at (x, y) usually defines the neighborhood about (x, y) (figure 3.7).

The simplest form of T is when the neighborhood is 1 x 1. In this case, g depends only on the value of f at (x, y) and T becomes a gray level transformation (mapping) function of the form s = T(r). For example, if T(r) has the form shown in the figure 3.8,

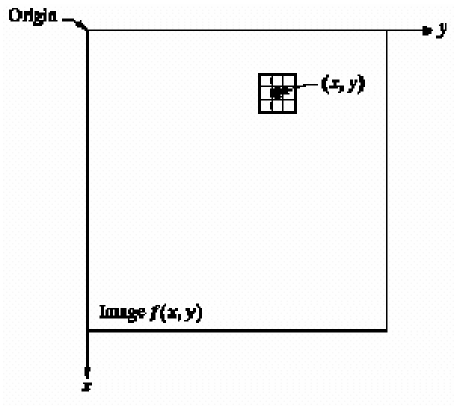


Figure 3.7 3 x 3 neighborhood about a point

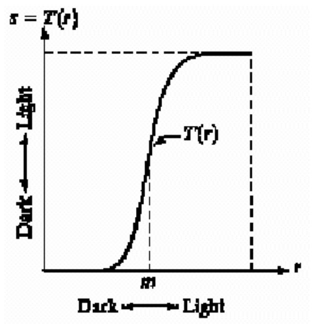


Figure 3.8 The transformation for higher contrast

the effect of this transformation is to produce an image of higher contrast than the original by darkening the levels below m and brightening the levels above m in the original image. This technique is known as contrast stretching. In the limiting case shown in the figure 3.9,

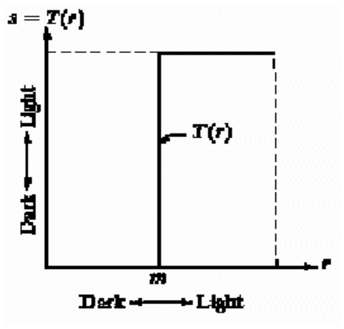


Figure 3.9 Limiting case transformations

T(r) produces a binary image. Because enhancement at any point in an image depends only on the gray level at that point, techniques in this category often are referred to as point processing (Gonzalez, 1992).

3.2.6.1 Image Enhancement by Point Processing

Image Negative: It is a useful process in some applications, such as displaying medical images or films. The negative of a digital image is obtained by using transformation function as shown in the below figure, L is the number of gray levels.

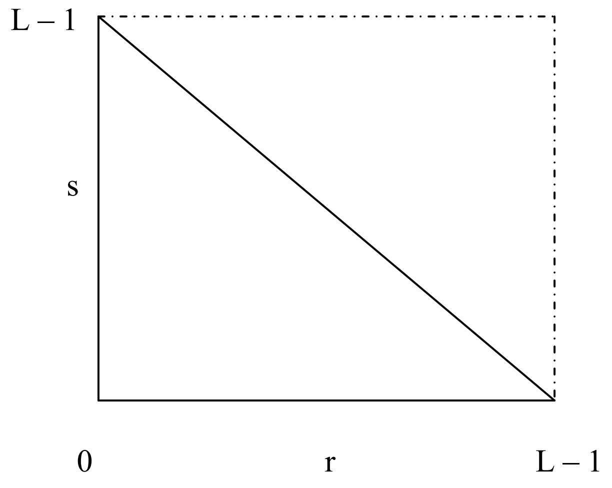


Figure3.10 Transformation function for image negative

Example of image negative is shown in figure 3.11.

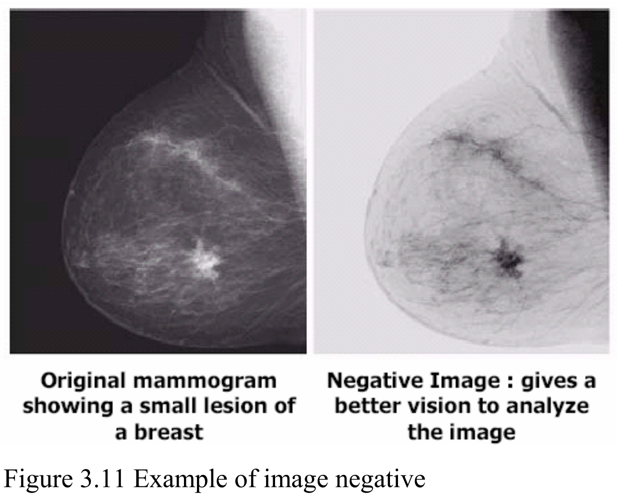

Original mammogram showing a small lesion of a breast
Negative Image : gives a better vision to analyze the image

Figure 3.11 Example of image negative

Contrast Stretching: Low-contrast images can result from poor illumination, lack of dynamic range in the imaging sensor, or wrong setting of a lens aperture during image acquisition. The idea behind contrast stretching is to increase dynamic range of the gray levels in the image being processed. The following figure shows a typical transformation used for contrast stretching.

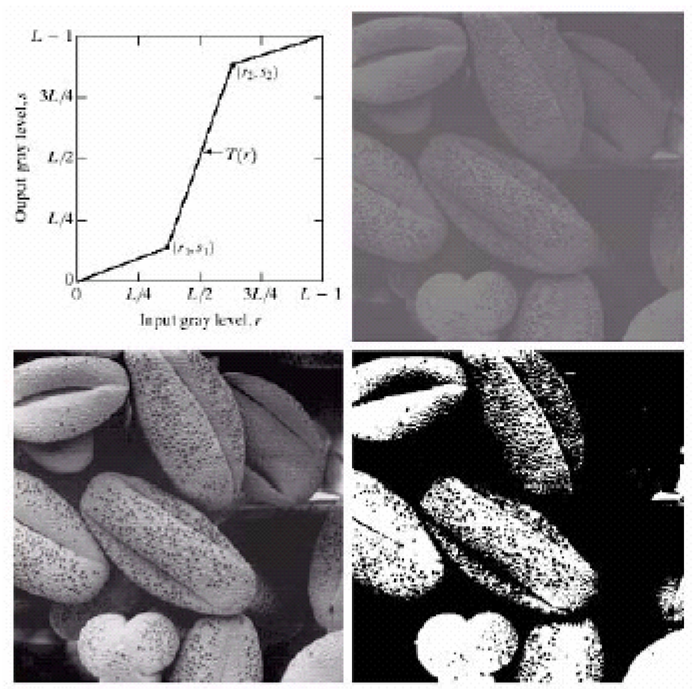


Figure 3.12 a) The transformation function for contrast stretching, b) a low contrast image, c) result of contrast stretching, d) result of thresholding.

Gray-level slicing: Highlighting a specific range of gray levels in an image often is desired. Applications include enhancing features such as masses of water in satellite imagery and enhancing flaws in x-ray images. One approach is to display a high value for all gray levels in the range of interest and a low value for all other gray levels. This transformation is shown in the figure 3.13.

The second approach, based on the transformation shown in the figure 3.12 (right hand side of the figure), brightens the desired range of gray levels but preserves the background in the image (Gonzalez, 1992).

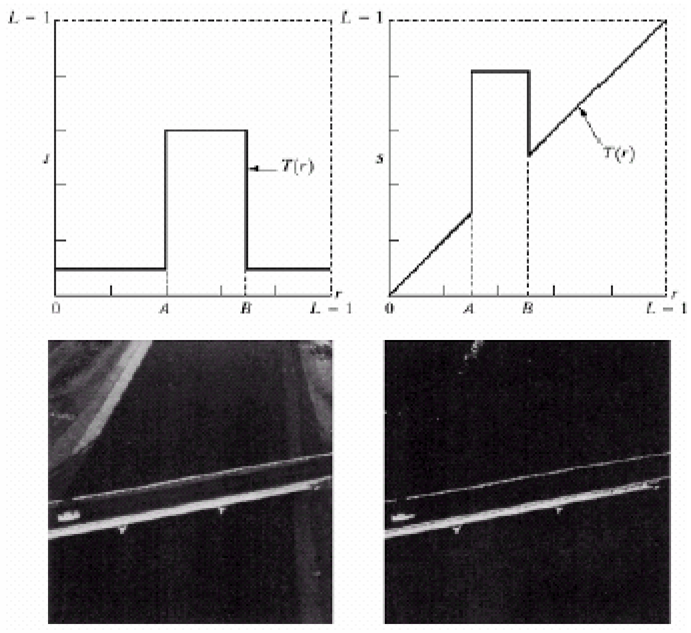


Figure 3.13 Gray level transformation

Histogram processing: Histogram of a digital image with gray levels in the range [0, L-1] is a discrete function $p(r_k) = n_k / n$, where r_k is the *k*th gray level, n_k is the number of pixels in the image with that gray level, n is the total number of pixels in the image, and k = 0,1,2,…,L-1. $p(r_k)$ gives an estimate of the probability of occurrence of gray level r_k .

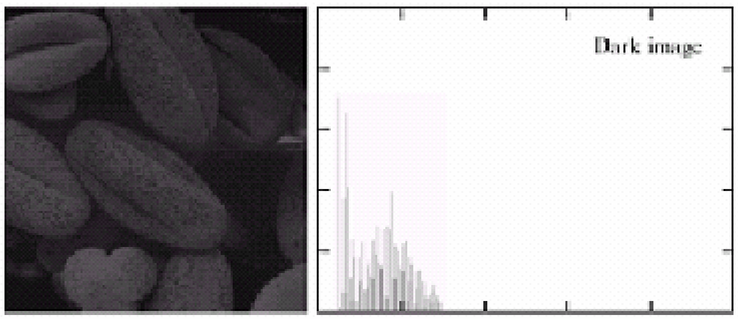


Figure 3.14 Histogram of the dark image

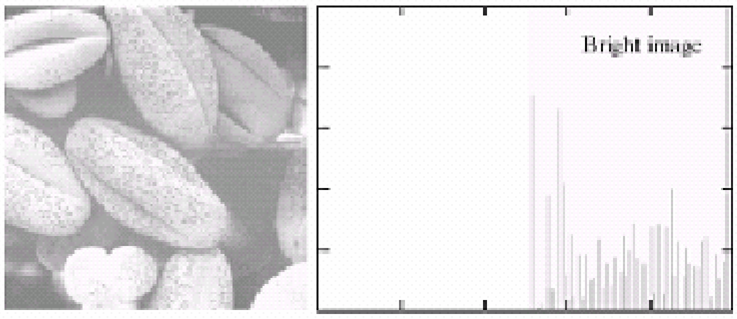


Figure 3.15 Histogram of the bright image

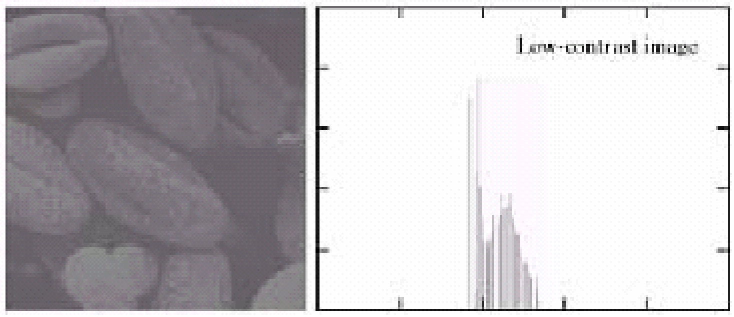


Figure 3.16 Histogram of the low-contrast image

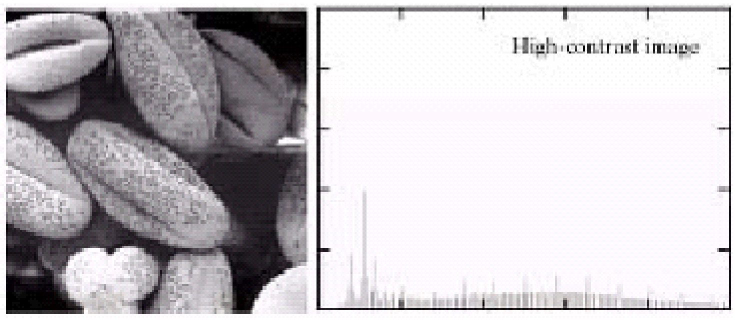


Figure 3.17 Histogram of the high-contrast image

Image subtraction: The difference between two images g(x, y) = f(x, y) – h(x, y) is obtained by computing the difference between all pairs of corresponding pixels from f and h. A classic application for enhancement is in an area of medical imaging called mask mode radiology. The example of mask mode radiography is shown in the figure 3.17 (Gonzalez, 1992).

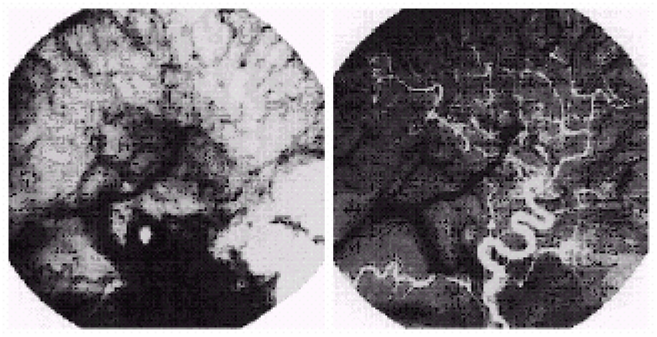


Figure 3.17 The left hand side of the figure is mask image, and the right hand side is an image with mask subtracted out.

3.2.7 Spatial Filtering

The use of spatial masks for image processing is called spatial filtering and the masks themselves are called spatial filters. Low-pass filters attenuate or eliminate high-frequency components in the Fourier domain while leaving low-frequencies untouched. High-frequency components characterize edges and other sharp details in an image, so the net effect of low-pass filtering is image blurring. Similarly, high-pass filters attenuate or eliminate low-frequency components. These components are responsible for the slowly varying characteristics of an image, such as overall contrast and average intensity; the net result of high-pass filtering is a reduction of these features, and sharpening of edges and other sharp details. Band-pass filtering removes selected frequency regions between low and high frequencies. These filters are used for image restoration.

In spatial filtering the basic approach is to sum products between the mask coefficients and the intensities of the pixel under the mask at a specific location in the image. The response of a 3 x 3 linear mask is

$$R = w_1 \cdot z_1 + w_2 \cdot z_2 + ... + w_9 \cdot z_9 \tag{3.8}$$

where w_i are the mask coefficients. The center of the mask is at location (x, y) in the image. The gray level of the pixel located at (x, y) is replaced by R. The mask is then moved to the next pixel location in the image and the process is repeated. Usual practice is to create a new image to store the values of R, instead of changing pixel values in place (Gonzalez, 1992).

3.2.7.1 Smoothing filters

Low-pass spatial filtering: A low-pass (smoothing) spatial filter has to have all positive coefficients. For a 3 x 3 spatial filter, the simplest arrangement would be a mask in which all coefficients have a value of 1. Since the response would be the sum of gray levels for nine pixels, R would be out of the valid gray level range. The solution is to scale the sum by dividing R by 9.

$$\frac{1}{9} \times \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$

Figure 3.18 3 x 3 spatial low-pass filter

In all cases the response R would be the average of all the pixels in the area of the mask. For this reason the use of the mask of the form shown in the figure above is often referred to as neighborhood averaging. The figure 3.19 shows the effect of low-pass filtering.

Median filtering: The smoothing method just discussed blurs edges and other sharp details. If the objective is to achieve noise reduction rather than blurring, an alternative approach is to use median filters. The gray level of each pixel is replaced by the median of the gray levels in a neighborhood of that pixel, instead of by the average. This method is particularly effective when the noise consists of strong, spike like components and the characteristic to be preserved is edge sharpness. Median filters are nonlinear. To perform median filtering in a neighborhood of a pixel, we first sort the values of the pixel and its neighbors, determine the median,

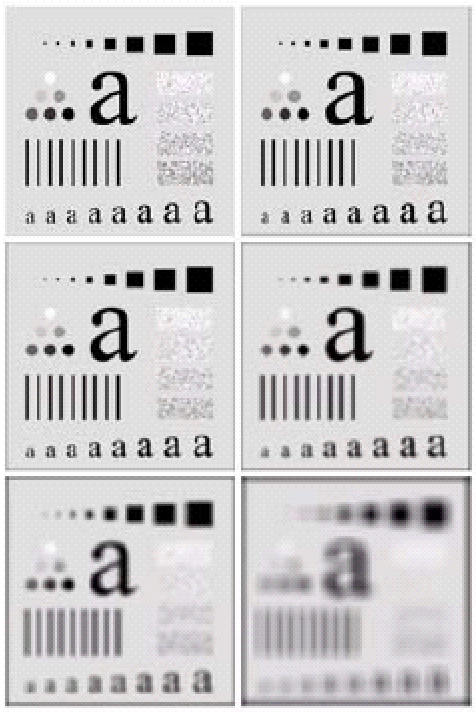


Figure 3.19 a) original image 500x500 pixels, b)…f) results of smoothing with square averaging filter masks of size n=3,5,9,15, and 35 respectively.

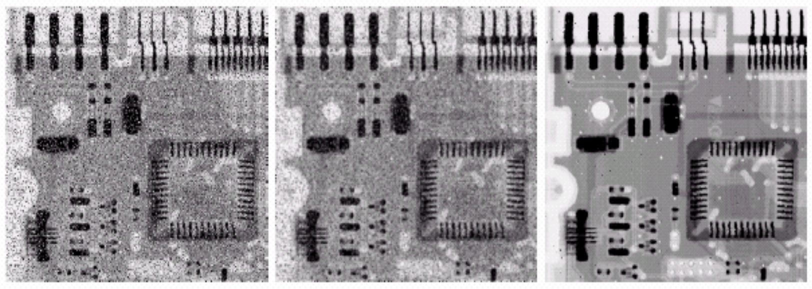
and assign this value to the pixel. For example, suppose that a 3 x 3 neighborhood has values,

10, 20, 20, 20, 15, 20, 20, 25, 100

assume that 100 is the gray level of the center pixel. These values are sorted as,

10, 15, 20, 20, 20, 20, 20, 25, 100

which results in a median of 20 (5*th* largest value). Then the gray level of the center pixel is replaced by 20. Median filtering forces points with distinct intensities to be more like their neighbors eliminating intensity spikes. The effect of median filtering is shown in figure 3.20 (Gonzalez, 1992).



a) b) c)

Figure 3.20 a) the original image, b) noise reduction with 3x3 averaging mask, c) median filtering.

3.2.7.2 Sharpening filters

Basic high-pass spatial filtering: The high pass spatial filter should have positive coefficients near its center and negative coefficients in the outer periphery. (figure 3.21)

1/9

-1	-1	-1
-1	8	-1
-1	-1	-1

Figure 3.21 A basic 3 x 3 sharpening high-pass filter

Note that the sum of the coefficients is 0. When the mask is over an area of constant or slowly varying gray level, the output of the mask is zero or very small. This filter eliminates the zero frequency term and reduces the average gray level value in the image to zero, reducing the global contrast of the image.

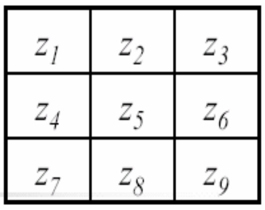
Derivative filters: As averaging is analogous to integration, and blurs detail in an image, differentiation is expected to have the opposite effect and thus sharpen an image. The most common method of differentiation in image processing applications is the gradient. The gradient of a function f(x, y) at coordinates (x, y) is defined as the vector.

$$\nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \dfrac{\partial f}{\partial x} \\ \dfrac{\partial f}{\partial y} \end{bmatrix} \qquad (3.9)$$

The magnitude of this vector:

$$\nabla f = mag(\nabla f) = \left[G_x{}^2 + G_y{}^2 \right]^{1/2} = \left[\left(\frac{\partial f}{\partial x} \right)^2 + \left(\frac{\partial f}{\partial y} \right)^2 \right]^{1/2} \qquad (3.10)$$

is the basis for various approaches to image differentiation. The equation given above can be approximated at image point Z_5 in a number of ways.



z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

Figure 3.22 A 3 x 3 image region

$G_x = (z_8 - z_5)$ and $G_y = (z_6 - z_5)$

Difference in x direction

Difference in y direction

$$\nabla f = \left[G_x{}^2 + G_y{}^2 \right]^{1/2} = \left[(z_8 - z_5)^2 + (z_6 - z_5)^2 \right]^{1/2} \qquad (3.11)$$

Instead of using squares and square roots, we can obtain similar results by using absolute values.

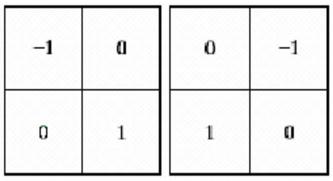
$$\nabla f \approx |z_8 - z_5| + |z_6 - z_5| \tag{3.12}$$

Another approach is to use cross differences:

$$\nabla f = \left[G_x{}^2 + G_y{}^2 \right]^{1/2} = \left[(z_9 - z_5)^2 + (z_8 - z_6) \right]^{1/2} \tag{3.13}$$

$$\nabla f \approx |z_9 - z_5| + |z_8 - z_6| \tag{3.14}$$

For example:



−1	0
0	1

0	−1
1	0

the above two masks are masks that can be used to implement cross differences. These masks are called *Robert's cross-gradient* operators.

An approximation to the magnitude of the gradient at point Z5, using 3 x 3 neighborhood is,

$$G_y = (z_7 + 2 \cdot z_8 + z_9) - (z_1 + 2 \cdot z_2 + z_3) \tag{3.15}$$

$$G_y = (z_3 + 2 \cdot z_6 + z_9) - (z_1 + 2 \cdot z_4 + z_7) \tag{3.16}$$

$$\nabla f \approx |G_x| + |G_y| \tag{3.17}$$

The pair of masks (called the *Sobel operators*) for approximating the magnitude of the gradient are shown in figure 3.23 (Gonzalez, 1992).

−1	−2	−1
0	0	0
1	2	1

−1	0	1
−2	0	2
−1	0	1

Figure 3.23 Sobel operators

3.2.8 Image Segmentation

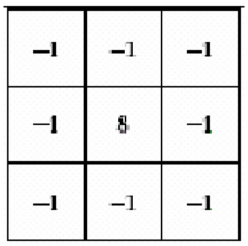
Image segmentation is the first step in image analysis, it subdivides an image into its constituent parts or objects. Subdivision is carried depending on the problem being solved. For example, in autonomous air-to-ground target acquisition applications, interest lies in identifying vehicles on a road. The first step is to segment the road from the image and then to segment the contents of the road down to objects that correspond to potential vehicles. There is no point in carrying segmentation below this scale and there is no need to attempt segmentation of image components that lie outside the boundaries of the road.

Segmentation algorithms generally are divided into two categories. In the first category, the approach is to partition an image based on abrupt changes in gray level (discontinuities). Detection of isolated points and detection of lines and edges in an image are in this category. The principal approaches in the second category are based on thresholding, region growing, and region splitting and merging (Gonzalez, 1992).

3.2.8.1 Detection of discontinuities

In practice, the most common way to look for discontinuities is to run a mask through the image (Gonzalez, 1992).

Point detection: Using the mask shown below, we say that a point has been detected at the location on which the mask is centered if $| R | > T$, where T is nonnegative threshold, and R is the response of the mask at that location.

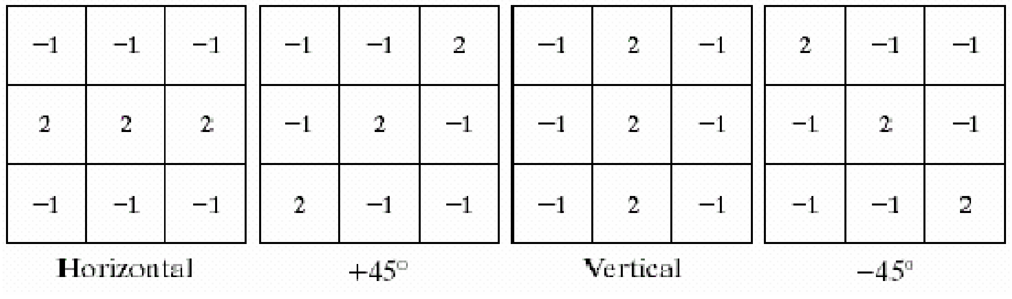


−1	−1	−1
−1	8	−1
−1	−1	−1

Figure 3.24 The mask for point detection

The idea is that the gray level of an isolated point will be quite different from the gray level of its neighbors. This mask is the same as the mask used for high frequency spatial filtering. The emphasis here is strictly on the detection of points.

Line detection:



−1	−1	−1
2	2	2
−1	−1	−1

Horizontal

−1	−1	2
−1	2	−1
2	−1	−1

+45°

−1	2	−1
−1	2	−1
−1	2	−1

Vertical

2	−1	−1
−1	2	−1
−1	−1	2

−45°

Figure 3.25 The masks used for line detection

If the first line is moved around the image, it responds more strongly to one pixel-thick lines oriented horizontally. When the line passes through the middle row of the mask, maximum response is obtained. The second mask responds best to lines oriented at 45^{o}. The third mask to vertical lines, and the fourth mask to lines in the -45^{o} direction. These directions can be established by noting that the preferred direction of each mask is weighted with a larger coefficient (that is 2 here) than other possible directions. Let R_1, R_2, R_3, and R_4 denote the responses of these masks given, from left to right. If, at a certain point in the image $| R_i | > | R_j |$, for all $j = i$, that point is said to be more likely associated with a line in the direction of mask i.

Edge detection: An edge is the boundary between two regions with relatively distinct gray level properties. The assumption, here, is that the regions in question are sufficiently homogeneous so that the transition between two regions can be determined on the basis of gray level discontinuities alone. The idea under the most edge detection techniques is the computation of a local derivative operator. The figure 3.25 illustrates this concept.

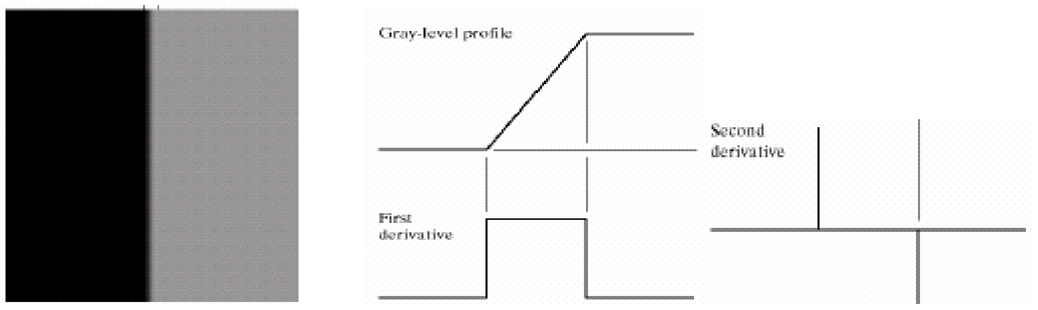


Figure 3.26 The result of first and second derivative application to the shown image

Note that an edge (transition from dark to light) is modeled as a smooth change of gray level. This model reflects the fact that edges in digital images are generally slightly blurred as a result of sampling.

The figure shows that the first derivative is positive at the leading edge of a transition, negative at the trailing edge, and zero in areas of constant gray level. The second derivative is positive for that part of the transition associated with the light side of the edge, and zero in areas of constant gray level. Zero crossing provide a powerful approach for locating edges in an image. Although this discussion has been limited to 1-D horizontal profile, a similar argument applies to an edge of any direction in an image. We simply define a profile perpendicular to the edge direction at any desired point and interpret the results as in the preceding discussion. The first derivative is obtained by using the magnitude of the gradient at any point. The second derivative is obtained by using the Laplacian (Gonzalez, 1992).

Gradient operators: In edge detection an important quantity is the magnitude of the gradient denoted by,

$$\nabla f = mag(\nabla f) = \left[G_x{}^2 + G_y{}^2\right]^{1/2} = \left[\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2\right]^{1/2}$$

or by a different approximation $\nabla f \approx |G_x| + |G_y|$

The direction of the gradient vector also is an important quantity.

$$\alpha(x, y) = \tan^{-1}\left(\frac{G_y}{G_x}\right) \tag{3.18}$$

Direction angle alfa is measured with respect to the x axis. Partial derivatives, G_x and G_y may be implemented in digital form in several ways as discussed in image enhancement, derivative filter section. However, the Sobel operators have the advantage of providing both a differencing and a smoothing effect. Because derivatives enhance noise, the smoothing effect is a particularly attractive feature of the Sobel operators. The example of Sobel gradient operators is given in the figure 3.27.

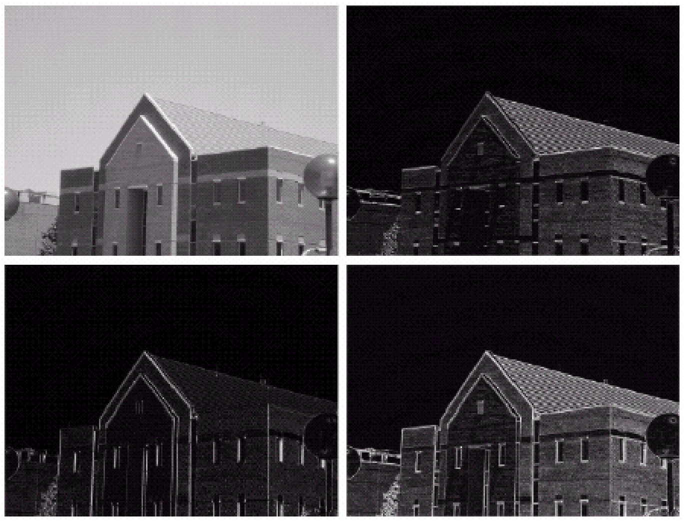


Figure 3.27 a) Original image, b) $|G_x|$, component of the gradient in the x-direction, c)$|G_y|$, component in the y direction, d) Gradient image $|G_x| + |G_y|$

Laplacian: The laplacian of a 2-D function f(x, y),

$$\nabla^2 f = \frac{\partial^2 f(x,y)}{\partial x^2} + \frac{\partial^2 f(x,y)}{\partial y^2} \tag{3.19}$$

$$\nabla^2 f = [f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1) - 4 \cdot f(x,y)] \tag{3.20}$$

The Laplacian may also be implemented in digital form in several ways. For a 3 x 3 image region, the most common practice is,

$$\nabla^2 f = 4 \cdot z_5 - (z_2 + z_4 + z_6 + z_8) \tag{3.21}$$

Because the Laplacian is a derivative, the sum of the coefficients has to be zero. So the response is zero when the point in question and its neighbors have the same value.

The Laplacian is seldom used in practice for edge detection for several reasons:

- Unacceptably sensitive to noise
- Produces double edges (when the magnitude of the Laplacian is thresholded to detect edges)
- Unable to detect edge direction.

A more general use of the Laplacian is in finding the location of edges using its zero crossing property (Gonzalez, 1992).

3.3 The Fourier Transform

3.3.1 The 1-Dimensional Fourier Transform

If f(x) is a continuous function of real variable x, the fourer transform of f(x) is given by,

$$F\{f(x)\} = F(u) = \int_{-\infty}^{\infty} f(x) \cdot e^{-j \cdot 2 \cdot \pi \cdot u \cdot x} dx \qquad \text{where } j = \sqrt{-1} \tag{3.22}$$

The inverse fourier transform of F(u) is,

$$F^{-1}\{F(u)\} = f(x) = \int_{-\infty}^{\infty} F(u) \cdot e^{j \cdot 2 \cdot \pi \cdot u \cdot x} du \tag{3.23}$$

These two equations are called the Fourier Transform pairs, and exists if f(x) is continuous and integrable and F(u) is integrable. We are considered with functions f(x) that are real. F(u) can be written in both cartesian and polar coordinates as follows (Gonzalez, 1992):

$$F(u) = R(u) + j \cdot I(u) \Rightarrow F(u) = |F(u)| \cdot e^{j\phi(u)} \tag{3.24}$$

$R(u)$ is real component, $I(u)$ is imaginary component.

Here,

$$|F(u)| = \left[R^2(u) + I^2(u)\right]^{1/2} \tag{3.25}$$

is Fourier spectrum and,

$$\phi(u) = \tan^{-1}\left[\frac{I(u)}{R(u)}\right] \tag{3.26}$$

is phase angle. The equation shown below,

$$P(u) = |F(u)|^2 = R^2(u) + I^2(u) \tag{3.27}$$

is commonly referred to as the power spectrum (spectral density) of f(x). u is called frequency variable.

3.3.2 The 2-Dimensional Fourier Transform

The Fourier Transform can be easily extended to a function f(x, y) of two variables, if f(x, y) is continuous and integrable and F(u, v) is integrable (Pratt, 1991).

$$F\{f(x,y)\} = F(u,v) = \iint f(x,y) \cdot e^{-j \cdot 2 \cdot \pi (u \cdot x + v \cdot y)} dx \cdot dy \tag{3.28}$$

$$F^{-1}\{F(u,v)\} = f(x,y) = \iint F(u,v) \cdot e^{j \cdot 2 \cdot \pi \cdot (u \cdot x + v \cdot y)} du \cdot dv \tag{3.29}$$

Here, u and v are the frequency variables.

Fourier spectrum is given by,

$$|F(u,v)| = \left[R^2(u,v) + I^2(u,v)\right]^{1/2} \tag{3.30}$$

and phase angle is given by,

$$\phi(u,v) = \tan^{-1}\left[\frac{I(u,v)}{R(u,v)}\right] \tag{3.31}$$

3.3.3 The Discrete Fourier Transform

Continuous function f(x) is discretized into a sequence

$$\{f(x_0), f(x_0 + \Delta x), f(x_0 + 2 \cdot \Delta x), \ldots\ldots\ldots\ldots, f(x_0 + [N-1] \cdot \Delta x\} \tag{3.32}$$

by taking N samples Δx units apart. It is possible to use x as a discrete variable by defining,

$$f(x) = f(x_0 + x \cdot \Delta x) \tag{3.33}$$

where x has the values 0,1,2,……,N-1.

The sequence $\{f(0), f(1), f(2), \ldots\ldots, f(N-1)\}$ denotes any N samples from a corresponding continuous function (Gonzalez, 1992).

The Discrete Fourier Transform pair of the sampled function is given by,

$$F(u) = \frac{1}{N}\sum_{x=0}^{N-1} f(x)\cdot e^{-j\cdot\frac{2\cdot\pi}{N}\cdot u\cdot x} \qquad \text{for u=0, 1, 2,.......,N-1} \tag{3.34}$$

$$f(x) = \sum_{u=0}^{N-1} F(u)\cdot e^{j\cdot\frac{2\cdot\pi}{N}\cdot u\cdot x} \qquad \text{for x=0, 1, 2,........,N-1} \tag{3.35}$$

The values u=0, 1, 2,……..,N-1 in the Discrete Fourier Transform correspond to samples of continuous transform at values $0, \Delta u, 2\cdot\Delta u, \ldots\ldots\ldots, (N-1)\cdot\Delta u$. In other words $F(u)$ represents $F(u\cdot\Delta u)$. The terms Δu and Δx are related by,

$$\Delta u = \frac{1}{N\cdot\Delta x} \tag{3.36}$$

In two variable case the Discrete Fourier Transform pair is,

$$F(u,v) = \frac{1}{M\cdot N}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)\cdot e^{-j\cdot 2\cdot\pi\left(\frac{u\cdot x}{M}+\frac{v\cdot y}{N}\right)} \tag{3.37}$$

$$f(x,y) = \sum_{u=0}^{M-1}\sum_{v=0}^{N-1} F(u,v)\cdot e^{j\cdot 2\cdot\pi\left(\frac{u\cdot x}{M}+\frac{v\cdot y}{N}\right)} \tag{3.38}$$

Sampling of a continuous function is now in 2-D grid, with divisions of width Δx and Δy in the x and y axis, respectively. The sampling increments in the spatial and frequency domains are related by,

$$\Delta u = \frac{1}{M\cdot\Delta x}\ ,\quad \Delta v = \frac{1}{N\cdot\Delta y} \tag{3.39}$$

3.3.4 The Fast Fourier Transform

The number of complex multiplications and additions required to implement 1-D Discrete Fourier Transform isproportional to N^2. The terms of $e^{-j\cdot 2\cdot \pi \cdot u\cdot x/N}$ can be computed once and stored in a table for all subsequent applications. For this reason multiplication of u by x is not considered a direct part of the implementation. Proper decomposition of the Discrete Fourier Transform equation can make the number of multiplication and addition operations proportional to $N \cdot \log_2 N$. The decomposition procedure is called the Fast Fourier Transform algorithm. The reduction in the operations represents a significant saving in computational effort. The Fast Fourier Transform approach offers a considerable computational advantage over direct implementation of the Fourier Transform, particularly when N is relatively large. N must be a power of 2. If not, zeros are appended to N samples extending it to M samples, where M is a power of 2 (Gonzalez, 1992).

3.3.4.1 The Inverse Fast Fourier Transform

If we take the complex conjugate of the inverse Discrete Fourier Transform equation and divide both sides by N,

$$f(x) = \sum_{u=0}^{N-1} F(u) \cdot e^{j\cdot \frac{2\cdot \pi}{N} \cdot u \cdot x} \Rightarrow \frac{1}{N} f^*(x) = \frac{1}{N} \sum_{u=0}^{N-1} F^*(u) \cdot e^{-j\cdot \frac{2\cdot \pi}{N} \cdot u \cdot x} \tag{3.40}$$

The equation now is in the form of forward Fourier Transform. Thus inputting $F^*(u)$ into an algorithm designed to compute the forward transform gives $f^*(x)/N$. Taking the complex conjugate and multiplying by N yields the desired result $f(x)$ (Gonzalez, 1992).

CHAPTER FOUR
SUPERVISED ORGANIZING MAP APPLICATION FOR SPEECH AND IMAGE RECOGNITION AND FUSION

The novel approach discussed in section 2.2 is applied for the fusion and recognition of image and speech. The used algorithm will be discussed in detail in this chapter.

4.1 Speech Processing in Matlab

The Matlab program development environment is used for the design of the proposed algorithm. As discussed in chapter three, the speech signal is variable signal, and processing the speech signal is one of the most difficult tasks.

The microphone is used for the acquisition of the speech signal. The speech signal is acquired in matlab by using the sound card of the computer and then sampled in order to discretize the speech signal. The sample frequency is 8 kHz, which is the Nyqist frequency of the maximum speech frequency in human speech signal, which is 4 kHz. Therefore according to Nyqist criterion, $f_s = 2 \cdot f_{\max} = 2 \cdot 4kHz = 8kHz$.

4.1.1 Filtering the Speech Signal

The microphone characteristic in the silent environment is shown in figure 4.1, and its Fourier spectrum is given in figure 4.2.

The higher amplitudes of the Fourier spectrum shown in figure 4.2 are at lower frequencies. Therefore these low frequency components must be eliminated by filtering the speech signal.

The human ear is sensible to sounds between the 20Hz – 20 kHz. The experimental results show that the daily human speech signal is between the frequencies of 100Hz to 2.5 kHz. Therefore the acquired speech signal can be filtered

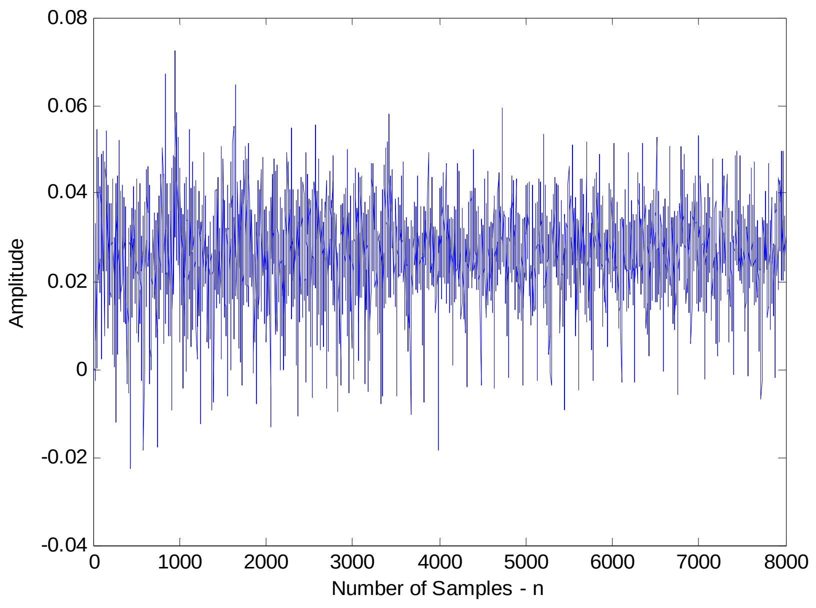


Figure 4.1 The signal acquired in silent environment

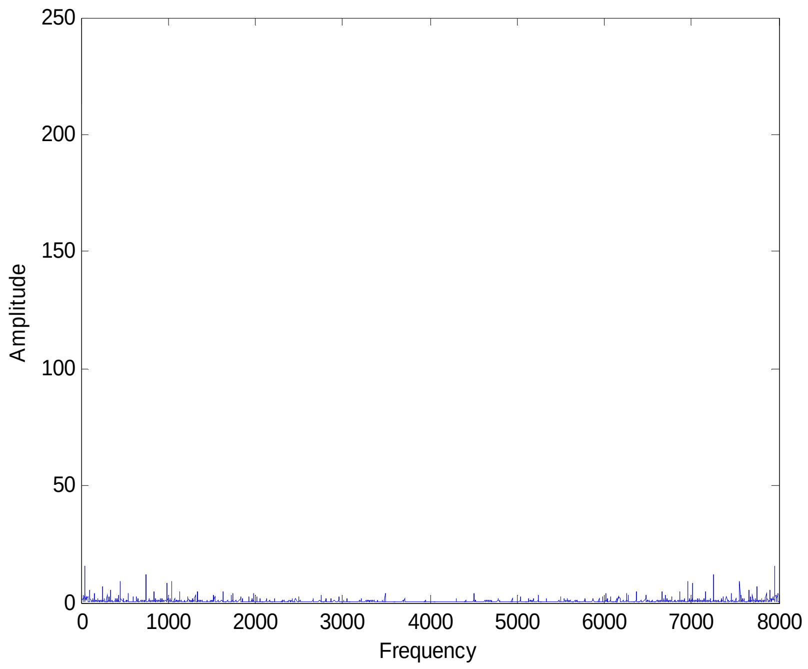


Figure 4.2 The Fourier spectrum of the signal shown in figure 4.1

by using the band pass filter. The microphone also has noise components below the 100 Hz frequency. Therefore it is a good choice to filter the signal with band pass filter. The band pass filter also eliminates the unnecessary high frequency components. The used band pass filter cut-off frequencies are $f_{c_1} = 100Hz$ and $f_{c_2} = 3kHz$. The filter transfer function is shown in figure 4.3.

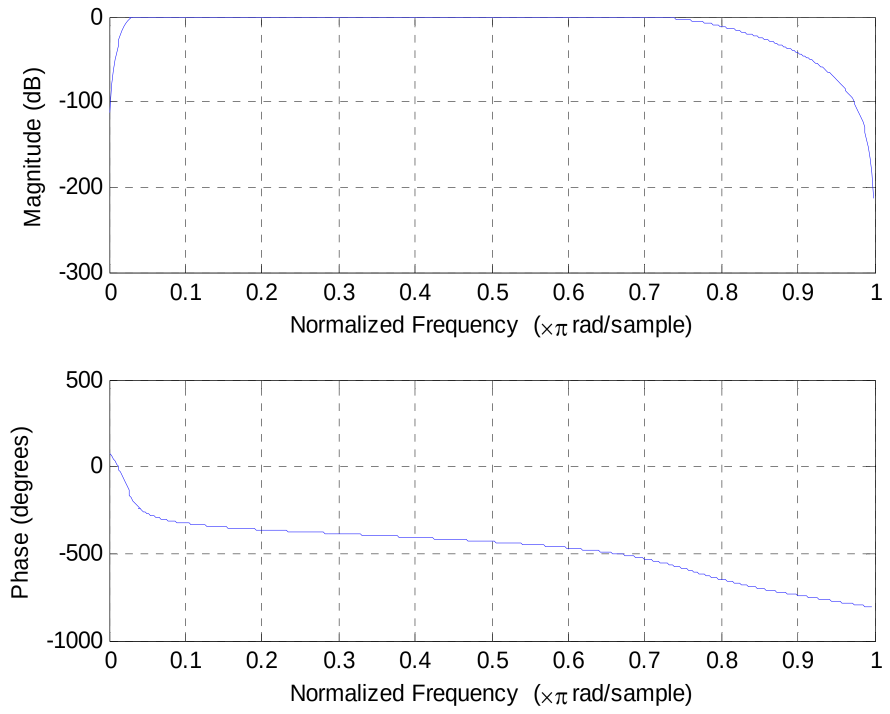


Figure 4.3 The transfer function of the used band pass filter

The maximum frequency in sampled speech signal is 4 kHz. Therefore the cut-off frequencies shown in figure 4.3 are normalized frequencies according to 4 kHz. The cut of frequency of the filter is defined to be the -3dB below the maximum magnitude. For the band pass filter shown in figure 4.3 the cut-off frequencies are $f_{c_1 normalized} = 100/4000 = 0.025$ and $f_{c_2 normalized} = 3000/4000 = 0.75$.

The speech signal must be normalized before processing. Because the speech signals amplitude makes big variations during the speech process. The amplitude of the speech signal depends on the closeness of the speaker to the microphone, and the speaker’s speech level. Therefore normalizing the speech signal decreases the effect of the amplitude variations of the speech signal, because we are interested in frequency components of a speech signal.

The figure 4.4 shows the normalized speech signal of letter ‘a’, the figure 4.5 shows its frequency spectrum, and figure 4.6 shows the frequency spectrum after filtering the speech signal with band pass filter which has the cut-off frequencies shown in figure 4.3.

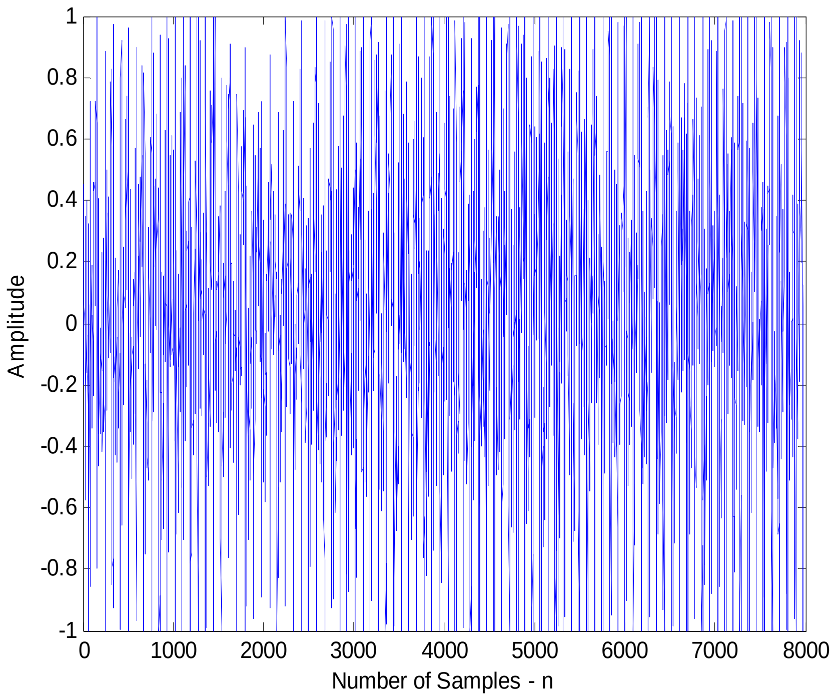


Figure 4.4 The normalized speech signal of letter ‘a’

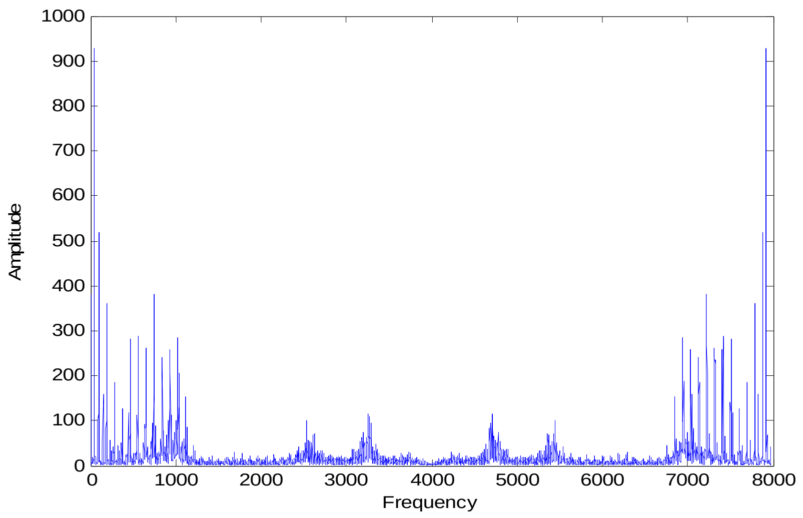


Figure 4.5 The frequency spectrum of signal shown in figure 4.4

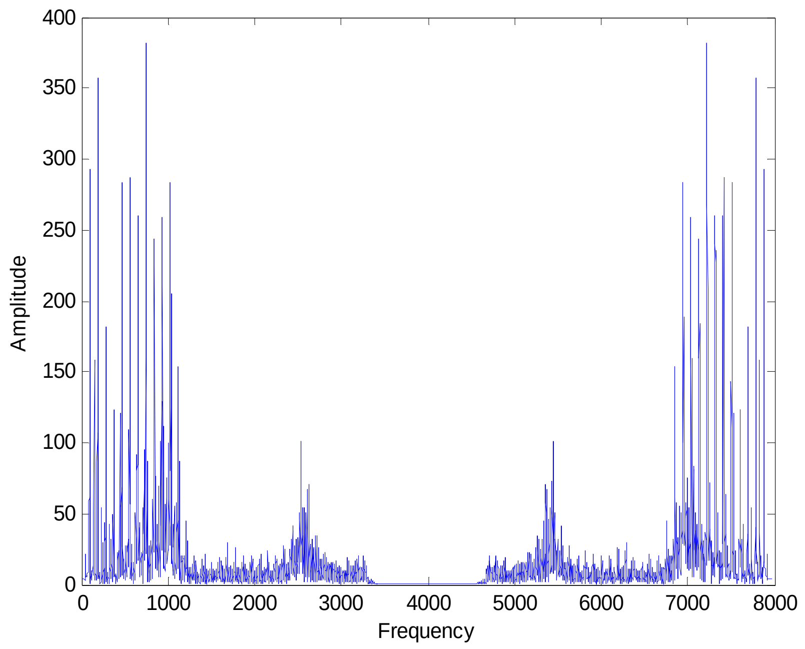


Figure 4.6 The frequency spectrum of signal shown in figure 4.4 after band pass filtering

4.1.2 Feature Extraction From Speech

Feature extraction is important task in speech recognition process. The features must be selected carefully in order to recognize the speech signal from its features. Because the speech signal is variable signal, its features are also variable. Therefore the used supervised organizing map algorithm plays very important role in speech recognition process.

The human brain processes the speech signal in a very complicated way, and the error in the brain for perception of the speech signal is almost zero. The learning process in the brain is also very complex task. But as discussed in chapter one, the main operation of the can be simulated. In the human ear perception of sound the receptors in the ear are divided into frequency bands. The small frequencies are divided into smaller bands. Every band is sensitive to the frequencies defined by the frequency bands. These properties of the human brain are tried to be simulated in this thesis work.

The information in the speech signal is carried by the frequency components of that signal. Therefore the Fourier Transform is used for the feature extraction from the speech signal. In the selection of the features, the means and variances of the Fourier spectrum are used. The small frequency components are divided into smaller bands. The total 94 features are used for the recognition of the speech signals. The acquired speech signal, after processing is transformed to the Fourier spectrum, and then the means and variance of the Fourier spectrum are determined as features. First the 94 selected features of speech signal are used as the weights for the first neuron class. Then the other speech signals and features corresponding to the same knowledge are used to train the supervised organizing map algorithm. In the following sections the used supervised organizing map algorithm and its use for image and speech fusion and recognition will be discussed in detail.

4.2 Image Processing in Matlab

Image processing takes very important role in image recognition and artificial intelligence applications. One of the most difficult tasks in image processing is the edge detection. Edge detection defines the edges of the object of interest. After detection of edges the features can be extracted by using the detected edge coordinates. But the edge detection is dependent on the background on which the object lies. This is very big problem in correct edge detection algorithms, because if the object color is same as the background color, or there is small difference, then the detection of the edges of the object is very difficult.

The human eye makes small vibrations when sensing the viewed objects C. Guzelis (lesson discussion, March 2005). In other words human brain takes the difference of the imaged sensed between the two vibration intervals. When simulating this property of the brain edge detection becomes easier. In order to simulate it, the camera can be connected to any motor, and the motor can be rotated to the clockwise and counter clockwise directions for very short time intervals. If the images are acquired between these short time intervals and extracted then the all edges in the sensed environment can be detected easily. But at this time the new problem arises. The recognition of the all detected edges is almost impossible task. The human brain recognizes all these edges without an error. But the human brain knows these detected edges from the early motion of these objects. As discussed in chapter one, the human brain learns objects from their motion and then saves all features corresponding to the same object as the one knowledge. The recognition of the all detected edges is very complex task and it is related with the attention system in the human brain.

The human brain vision system is very difficult task to be simulated by the computer program because of the reasons discussed in the above paragraph. Therefore the dinosaur vision system is tried to be simulated. The dinosaur cannot see the objects which are not moving C. Guzelis (lesson discussion, March 2005). This means that the dinosaur eye does not make small vibrations, therefore the object must move for dinosaur to detect it.

4.2.1 The Image Subtraction

The image subtraction is easy task to be performed. The Matlab image acquisition toolbox is used for the acquisition of images. As an imaging sensor, the standard webcam is used, and it is connected to the computer via the USB port. The resolution of the webcam is 240 x 320.

In the application of image subtraction, the images are acquired between the small time intervals. After the extraction of the images the edges of the moving objects are determined easily. The figure 4.7 shows the detected edges of the glass of water.

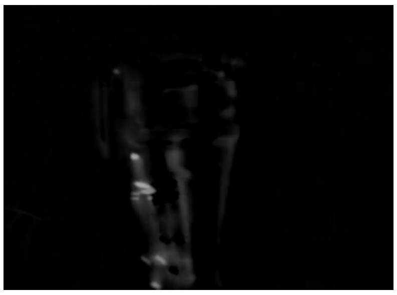


Figure 4.7 Detected edges of the glass

4.2.2 Filtering the Image

In chapter three the filtering techniques are discussed. The Matlab image processing toolbox has the built in functions which make its implementation easier. The median filtering is used in the first process of image. The median filtering is applied before the image subtraction in order to remove the noise caused by improper detection of pixel values.

After the detection of the edges of the object the high pass filtering is used to filter the low frequency components of the image, because we are interested in high frequency components of the image. In the detection of edges of the moving object by using image subtraction, the more information is at high frequency components. The transfer function of the two dimensional high pass filtering is shown in figure 4.8.

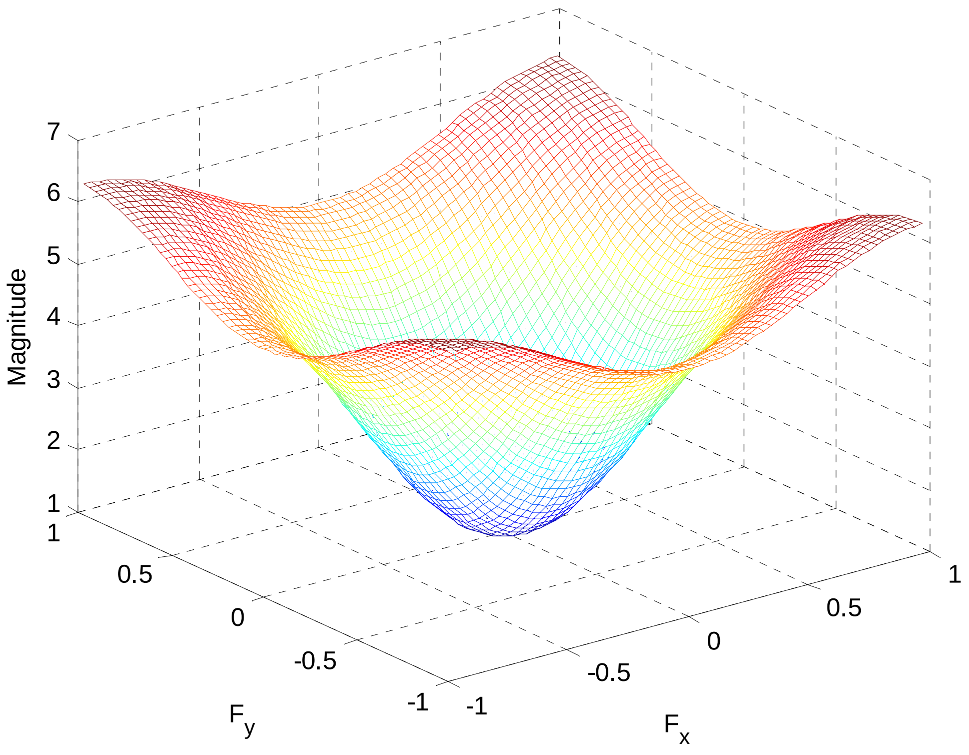


Figure 4.8 The transfer function of two dimensional high pass filter

The filtered output of the image shown in figure 4.7 by using the high pass filter, which has the transfer function shown in figure 4.8, is shown in figure 4.9.

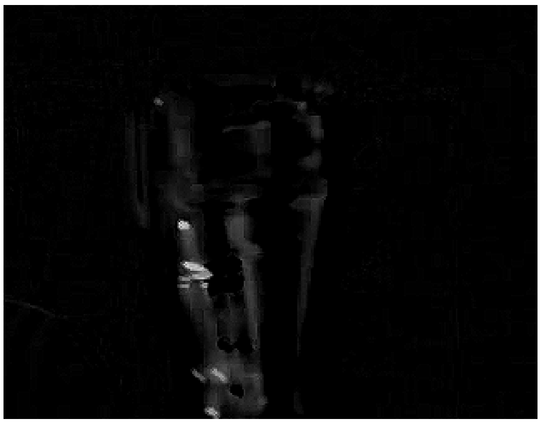


Figure 4.9 The result of filtering the image shown in figure 4.7 by using the two dimensional high pass filter

After filtering the image, the two dimensional Fast Fourier Transform algorithm is used to determine the frequency components of the image. The Fourier spectrum of the after filtering is shown in figure 4.10.

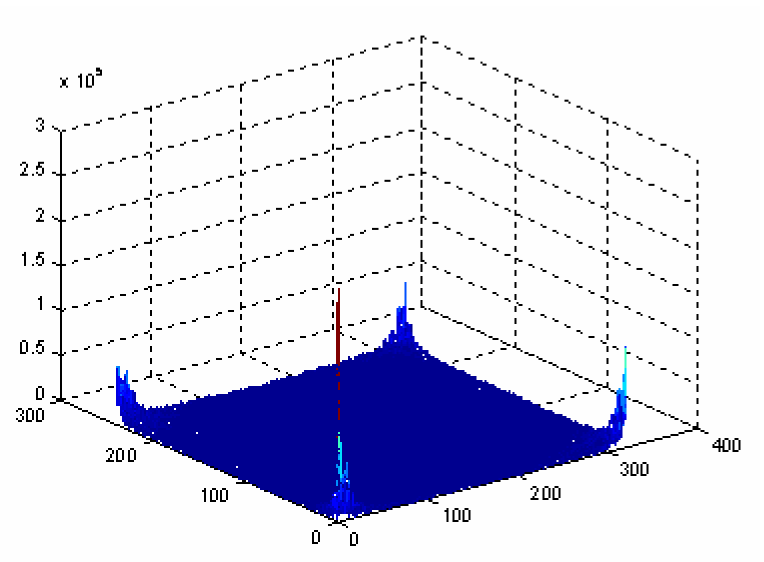


Figure 4.10 The Fourier spectrum of the filtered image shown in 4.9

4.2.3 Feature Extraction from Image

As in the feature extraction from speech, the Fourier Transform coefficients are used as features for the acquired image data. The two dimensional Fourier Transform is used to determine the frequency variations in the image.

The Fourier Transform coefficients in x and y directions are divided into equal bands, and the means and variances of the frequency bands are used as features for the image. The total 48 features are used for the recognition of the acquired image. The fusion of image and speech features and algorithm to update the neuron weights of the supervised organizing map will be discussed in the following section.

4.3 The Supervised Organizing Map Algorithm

The Supervised Organizing Map algorithm is discussed in chapter two. In this chapter the adaptation of the Supervised Organizing Map algorithm to the image and sound recognition and fusion will be discussed.

The Supervised Organizing Map algorithm is a new algorithm, which suggests the image and speech recognition processes closer to the processes performed by the human brain. The disadvantage of the algorithm is the long learning time, because the new knowledge and the recognized objects features are learnt by the aid of the user (teacher) at every input samples. But if the learning time of the human brain is considered at the first stages of the learning process, the proposed algorithm’s learning time is not long.

At the beginning of the algorithm when there is no knowledge present, the algorithm requires image and speech inputs to be inputted simultaneously to the algorithm. If not algorithm waits until the image and sound input is acquired simultaneously. After the simultaneous image and speech input, the 94 features of the speech signal and 48 features of the image are saved in the memory field. This corresponds to the first knowledge for the algorithm. For example, if the image of glass is shown and the word “glass” is spoken by the user, the features of the glass image and “glass” word are saved. The algorithm repeats itself in a forever loop. Therefore at the second execution of the algorithm, there will be three cases:

- *Case1:* If only image input is present to the algorithm, the algorithm finds the closest neuron to the determined features, and then shows the image and outputs the speech signal for the teacher to define whether it is correct knowledge or not. If the user defines that it is correct knowledge, the weights of the neuron class, which are the features of the known image are updated according to the update equation given below.

$$w_k(t+1) = w_k(t) + \alpha \cdot (x(t) - w_k(t)) \tag{4.1}$$

where α is the learning rate.

- *Case2:* If only sound input is present to the algorithm, the algorithm finds the closest neuron to the determined features, and then shows the image and outputs the speech signal for the teacher to define whether it is correct knowledge or not. If the user defines that it is correct knowledge, the weights of the neuron class, which are the features of the known sound are updated according to the update equation given below.

$$W_k(t+1) = W_k(t) + \alpha \cdot (x(t) - W_k(t)) \tag{4.2}$$

where α is the learning rate.

- *Case3:* If image and sound inputs are present to the algorithm, the algorithm finds the closest neurons to the determined features of image and speech, and then shows the image and outputs the speech signal for the teacher to define whether it is correct knowledge or not. If the user defines that it is correct knowledge, the weights of the neuron classes, which are the features of the known sound and image are updated according to the update equations given below.

$$w_k(t+1) = w_k(t) + \alpha \cdot (x(t) - w_k(t))$$

$$W_k(t+1) = W_k(t) + \alpha \cdot (x(t) - W_k(t))$$

The w_k corresponds to weights of the neurons used to classify the image inputs, and capital W_k corresponds to weights of the neurons used to classify the speech inputs.

In the Case1 and Case2, if the user defines that it is not the correct knowledge, then algorithm continuous its operation from the beginning without making any change to the weights of the neurons. In Case3, if the user defines that it is not the correct knowledge, then the features of the image and speech are saved as a new knowledge. The features of image and speech are used as the initial weights of the new neuron class.

The aim of the algorithm is to classify the speaker independent speech signals, which correspond to the same knowledge, in one neuron class, and also to classify the various image inputs corresponding to the same knowledge in one neuron class.

The experimental results show that the proposed algorithm is applicable on the image and speech recognition tasks. The effectiveness of the Supervised Organizing Map will be discussed in chapter five.

CHAPTER FIVE
CONCLUSION

The designed Supervised Organizing Map algorithm is trained by the samples for sound taken from the different speakers, and for image taken from various environments. After the enough training iterations (the number of training iterations is not defined mathematically, it depends on the performance of the network), the weights of the neurons tend to be placed in the middle of the various input samples corresponding to the same knowledge. The system performance is observed to be %99.5 when one speaker is present and the illumination in the environment is same for acquisition of images. In the case of two or more speakers and variable illuminations in the environment, the system performance is observed to be more than %80. These results are obtained by selecting the learning rate α to be 0.3. The neuron classes are defined by 94 weights for speech and 48 weights for image. Therefore it is not possible to plot the system performance graphically. But, by selecting the two weights for neuron classes and two features of input samples, the movement of the neuron classes according to the input samples for speech input is shown in figure 5.1

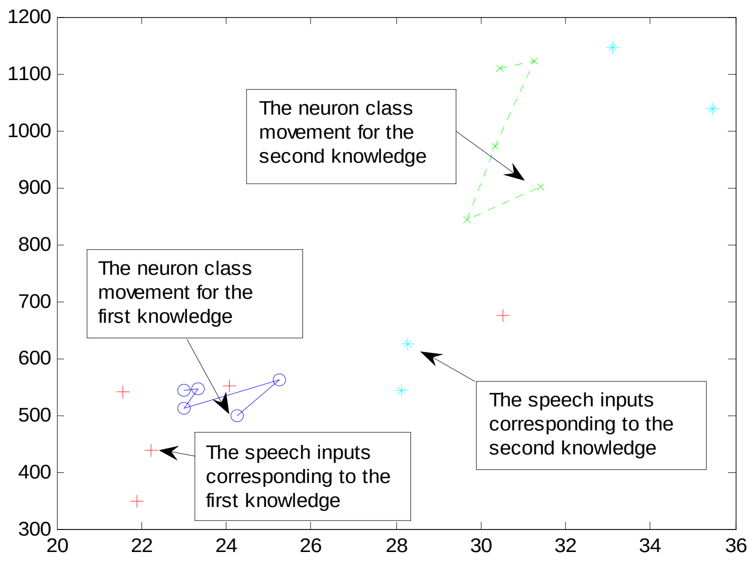


Figure 5.1 The movements of the neuron classes according to the input samples

The designed algorithm can be used to train the network for various inputs of the same knowledge. After the training process, the final weights of the neurons can be used for recognition of image and sound, or for any other desired classification problem.

The main difference between the Self Organizing Map and Supervised Organizing Map can be seen from figure 5.1. The input which is shown by the + sign and red color, would be classified to the second knowledge in Self Organizing Map algorithm, because the features are closer to the second neuron class. But in Supervised Organizing Map the features of the input is defined by the user to which neuron class it belongs. Therefore the neuron class corresponding to the first knowledge is given chance to update its weight vector.

It can be seen from figure 5.1 the inputs are not linearly separable, but, the figure is shown only by using the two weights of the neurons. In system there are 94 weights for neurons classifying the speeches, and 48 weights for neurons classifying the images. Therefore the feature input vectors are expected to be linearly separable.

In the future works, the better features can be selected for classifying the speeches and images, such as features discussed in the chapter one. The human brain attention system is very complex, but understanding the main idea behind it can be helpful for developing the new algorithms for artificial intelligence applications.

REFERENCES

Ben-Yacoub, S., Abdeljaoued, Y., & Mayoraz, E. (1999). Fusion of face and speech data for person identity verification. *IEEE Transactions on Neural Networks,* 10 (5), 1065-1074.

Crick, F. (2003). *Şaşırtan Varsayım* (10. Baskı). (Say, S., Çev.). Türkiye Bilimsel ve Teknik Araştırmaları Kurumu Yayınları. (Orijinal çalışma basım tarihi 1990).

Gonzalez, R. C., & Woods, R. E. (1992). *Digital Image Processing.* Addison Wesley.

Grossberg, S. (1982). *Studies of Mind and Brain*. Boston, Reidel.

Hebb, D. (1949). *The organization of behavior*. New York, John Wiley & Sons.

Ichiki, H., Hagiwara, M., & Nakagawa, M. (1993). Kohonen Feature Maps as a Supervised Learning Machine. *IEEE International Conference on Neural Networks*, 3, 1944-1948.

Kasabov, N. (1996). *The Foundations of Neural Networks and Fuzzy Systems, and Knowledge Engineering.* Massachusetts: The M.I.T. Press.

Khaparde, S. A., & Gandhi, H. (1993). Use of Kohonen's Self Organizing Network as a Pre-Quantizer. *Proceedings of the IEEE,* 5, 967-971.

Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43, 59-69.

Kohonen, T. (1990). The self-organizing map. *Proceedings of the IEEE* 78, 1464-1497.

Kohonen T. (1993). Physiological interpretation of the self-organizing map algorithm. *Neural Networks* 6, 895-905.

Kosko, B. (1988). Feedback stability and unsupervised learning. *In Proceedings of the Second IEEE International Conference on Neural Networks*, 1, 141-152.

Kosko, B. (1990). Unsupervised learning in noise. *IEEE Transactions on Neural Networks* 1, 44-57.

Morgan, D., Scofield, C. (1991). *Neural networks and speech processing*. Boston, Kluwer Academic.

Painter, T., & Spanias, A. (2000). Perceptual Coding of Digital Audio. *Proceedings of the IEEE*, 88 (4), 451-513.

Penrose, R. (2004). *Bilgisayar ve Zeka* (12. Baskı). (Dereli, T., Çev.). Türkiye Bilimsel ve Teknik Araştırmaları Kurumu Yayınları. (Orijinal çalışma basım tarihi 1989).

Pratt, W.K. (1991). *Digital Image Processing.* John Willey & Sons.

APPENDIX
MATLAB CODES

```
%Analog Signal Input
ai = analoginput('winsound');
chan=addchannel(ai,1);
duration=2;
set(ai,'SampleRate',8000);
ActualRate = get(ai,'SampleRate')
set(ai,'Timeout',2.14748e+006)
set(ai,'SamplesPerTrigger',8000)
set(ai,'TriggerChannel',chan)
set(ai,'TriggerType','Software')
set(ai,'TriggerCondition','Rising')
set(ai,'TriggerConditionValue',0.5)
start(ai)
data1 = getdata(ai,8000);

%Normalization of Speech Signal
wet=find(data1>0.01);
actsignal= data1(wet(1):wet(length(wet)));
normlzd=max(actsignal);
normsignal=actsignal/normlzd;
data=normsignal;

%Band-Pass Filtering
w=0;
z=0;
had=0;
n=5;
Wn=[100/4000 3000/4000];
[w,z]=butter(n,Wn);
```

```
had=filter(w,z,data);

%Fourier Transform
fourier_transform=fft(had,8000);
fourier_trans=abs(fourier_transform);

%Feature Extraction
feature_extract=reshape((fourier_trans)',80,100);
feature1=reshape(feature_extract(1,:),5,20);
feature2=reshape(feature_extract(2,:),5,20);
feature3=reshape(feature_extract(3,:),5,20);
feature4=reshape(feature_extract(4,:),5,20);
feature5=reshape(feature_extract(5,:),5,20);
feature6=reshape(feature_extract(6,:),2,50);
feature7=reshape(feature_extract(7,:),2,50);
feature8=reshape(feature_extract(8,:),2,50);
feature9=reshape(feature_extract(9,:),2,50);
feature10=reshape(feature_extract(10,:),2,50);

means1=mean(feature1');
means2=mean(feature2');
means3=mean(feature3');
means4=mean(feature4');
means5=mean(feature5');
means6=mean(feature6');
means7=mean(feature7');
means8=mean(feature8');
means9=mean(feature9');
means10=mean(feature10');
means11=mean(feature_extract((11:22),:)');

vars1=var(feature1');
```

```
vars2=var(feature2');
vars3=var(feature3');
vars4=var(feature4');
vars5=var(feature5');
vars6=var(feature6');
vars7=var(feature7');
vars8=var(feature8');
vars9=var(feature9');
vars10=var(feature10');
vars11=var(feature_extract((11:22),:)');

%Image Acquisition
image=videoinput('winvideo',1);
image_data1=getsnapshot(image);
test_count=0;
for i1=1:10000,
   test_count=test_count+1;
end
image_data2=getsnapshot(image);

%RGB to Gray and Median Filtering
im11=rgb2gray(image_data1);
im1=medfilt2(im11);
im22=rgb2gray(image_data2);
im2=medfilt2(im22);
image_data=im2-im1;

%Filtering the Image
h=fspecial('unsharp');
filtered_image=imfilter(image_data,h);
```

```
%Two Dimensional Fourier Transform
fft_image=fft2(double(filtered_image));
fft_image=abs(fft_image);

%Feature Extraction
image_feature1=fft_image(1:10,:);
temp1=sum(image_feature1);
im_mean1=mean(temp1);
im_var1=var(temp1);

image_feature2=fft_image(11:20,:);
temp2=sum(image_feature2);
im_mean2=mean(temp2);
im_var2=var(temp2);

image_feature3=fft_image(21:30,:);
temp3=sum(image_feature3);
im_mean3=mean(temp3);
im_var3=var(temp3);

image_feature4=fft_image(31:40,:);
temp4=sum(image_feature4);
im_mean4=mean(temp4);
im_var4=var(temp4);

image_feature5=fft_image(41:50,:);
temp5=sum(image_feature5);
im_mean5=mean(temp5);
im_var5=var(temp5);

image_feature6=fft_image(51:60,:);
temp6=sum(image_feature6);
```

```
im_mean6=mean(temp6);
im_var6=var(temp6);

image_feature7=fft_image(61:70,:);
temp7=sum(image_feature7);
im_mean7=mean(temp7);
im_var7=var(temp7);

image_feature8=fft_image(71:80,:);
temp8=sum(image_feature8);
im_mean8=mean(temp8);
im_var8=var(temp8);

image_feature9=fft_image(81:90,:);
temp9=sum(image_feature9);
im_mean9=mean(temp9);
im_var9=var(temp9);

image_feature10=fft_image(91:100,:);
temp10=sum(image_feature10);
im_mean10=mean(temp10);
im_var10=var(temp10);

image_feature11=fft_image(101:110,:);
temp11=sum(image_feature11);
im_mean11=mean(temp11);
im_var11=var(temp11);

image_feature12=fft_image(111:120,:);
temp12=sum(image_feature12);
im_mean12=mean(temp12);
im_var12=var(temp12);
```

```
image_feature13=fft_image(121:130,:);
temp13=sum(image_feature13);
im_mean13=mean(temp13);
im_var13=var(temp13);

image_feature14=fft_image(131:140,:);
temp14=sum(image_feature14);
im_mean14=mean(temp14);
im_var14=var(temp14);

image_feature15=fft_image(141:150,:);
temp15=sum(image_feature15);
im_mean15=mean(temp15);
im_var15=var(temp15);

image_feature16=fft_image(151:160,:);
temp16=sum(image_feature16);
im_mean16=mean(temp16);
im_var16=var(temp16);

image_feature17=fft_image(161:170,:);
temp17=sum(image_feature17);
im_mean17=mean(temp17);
im_var17=var(temp17);

image_feature18=fft_image(171:180,:);
temp18=sum(image_feature18);
im_mean18=mean(temp18);
im_var18=var(temp18);

image_feature19=fft_image(181:190,:);
```

```
temp19=sum(image_feature19);
im_mean19=mean(temp19);
im_var19=var(temp19);

image_feature20=fft_image(191:200,:);
temp20=sum(image_feature20);
im_mean20=mean(temp20);
im_var20=var(temp20);

image_feature21=fft_image(201:210,:);
temp21=sum(image_feature21);
im_mean21=mean(temp21);
im_var21=var(temp21);

image_feature22=fft_image(211:220,:);
temp22=sum(image_feature22);
im_mean22=mean(temp22);
im_var22=var(temp22);

image_feature23=fft_image(221:230,:);
temp23=sum(image_feature23);
im_mean23=mean(temp23);
im_var23=var(temp23);

image_feature24=fft_image(231:240,:);
temp24=sum(image_feature24);
im_mean24=mean(temp24);
im_var24=var(temp24);

%Minimum Distance and Winning Neuron Determination (Two Knowledge Case)
for j=1:10,
    if (sound_features(j,1)~=0)
```

```
        thought_temp(j,:)=sound_features(j,:)-sound_feature;
        thought_itemp(j,:)=image_features(j,:)-image_feature;
        end
    end
    thought_temp=abs(thought_temp);
    thought_itemp=abs(thought_itemp);
    thought_temp=sum(thought_temp');
    thought_itemp=sum(thought_itemp');
    thought_imtemp=thought_temp+thought_itemp;
    ok1=0;
    know1=length(thought_imtemp);
for p=1:know1,
    if (ok1==0)
    thought1=min(thought_imtemp);
    thought=find(thought_imtemp==thought1);

    if (thought==1)
        close all;
        load sounds_data1
        speak=ifft(fourier_trans1);
        sound(speak)
        im1=imread('image1.jpg');
        imshow(im1)
        reply = input('Is it Right? If yes press-1, If no press-2:\n');
        if (reply==1)
            sound_features(1,:)=sound_features(1,:)+0.3*(sound_feature-
sound_features(1,:));
            image_features(1,:)=image_features(1,:)+0.3*(image_feature-
image_features(1,:));
            knowledge=1;
            ok1=1;
        end
        if (reply==2)
            thought_imtemp(1)=1000000000000;
        end
    end

    thought1=min(thought_imtemp);
    thought=find(thought_imtemp==thought1);

    if (thought==2)
        close all;
        load sounds_data2
        speak=ifft(fourier_trans2);
        sound(speak)
        im2=imread('image2.jpg');
        imshow(im2)
        reply = input('Is it Right? If yes press-1, If no press-2:\n');
        if (reply==1)
            sound_features(2,:)=sound_features(2,:)+0.3*(sound_feature-
sound_features(2,:));
            image_features(2,:)=image_features(2,:)+0.3*(image_feature-
image_features(2,:));
            knowledge=1;
            ok1=1;
        end
        if (reply==2)
            thought_imtemp(2)=1000000000000;
        end
    end

%Correct Knowledge Determination (Two Knowledge Case)
if (knowledge==0)
        knowledge_no=length(thought_imtemp);
        if (knowledge_no==1)
            sound_features(2,:)=sound_feature;
            image_features(2,:)=image_feature;
            fourier_trans2=fourier_transform;
            save sounds_data2 fourier_trans2
            imwrite(image_data,'image2.jpg');
        end

        if (knowledge_no==2)
            sound_features(3,:)=sound_feature;
            image_features(3,:)=image_feature;
            fourier_trans3=fourier_transform;
            save sounds_data3 fourier_trans3
            imwrite(image_data,'image3.jpg');
        end
end
```