CHAPTER 3
THEORETICAL BACKGROUND

This chapter establishes a theoretical basis for the methods and strategies presented in subsequent chapters.

3.1. Image Segmentation

Image segmentation is one of the most complicated tasks in image processing, which is regarded as a significant essential operation for useful analysis and interpretation of obtained image. Image segmentation is a technique which is used for segmenting digital image into multiple regions or clusters. Each region of the image is comprised as sets of pixels. Image segmentation facilitates and alters the representation of an image by converting the image into something that is more meaningful and easier to examine. Locating the objects of interest and boundaries like lines, curve in an image is the typical use of image segmentation [35]. Several image, video and computer vision applications use image segmentation as an important processing step. Far-reaching research has been carried out in developing several diverse methods and algorithms for image segmentation, yet it is still hard to appraise if one algorithm obtains more precise segmentations than another, whether it be for a specific image or set of images, or more broadly, for an entire class of images [36].

Image segmentation is actually done as a pre-processing step in a number of image understanding applications, for example in certain medical imaging systems. The intention of segmentation algorithm is to partition the image into semantically momentous regions, or objects, to be identified by subsequent processing steps [37, 38]. It is, however, well known that semantically important regions are identified in an image at various scales of analysis. For a high resolution aerial image, we can recognize individual trees or plants at finer scales but, we can’t
cover fields at coarse scales. In single-scale segmentation algorithm, the parameters and thresholds should be tuned at the right scale of analysis. But it is often impractical to decide on the exact scale of analysis in advance, because diverse types of images require diverse scales of analysis, and also, in several cases important objects appear at different scales of analysis in the same image.

3.1.1. Anisotropic Diffusion

For acquiring image smoothing and segmentation, several models of linear and nonlinear diffusion have been suggested in literature. Several researchers have also suggested nonlinear anisotropic diffusion [39], [40]. Image Smoothing is developed as a diffusive process, which is attenuated or blocked at edges by selecting locally adaptive diffusion strengths. Higher levels of image processing employ anisotropic diffusion as a preprocessing step. It smoothes image interiors to highlight the boundaries for segmentation; also it removes forged feature to progress the response of edge detection algorithms and also proves competent at eliminating noise from images [41]. But low frequency artifacts are hard to dispel without over-processing the image are likely to be produced by relaxation procedures that establish anisotropic diffusion. In image processing, anisotropic diffusion discretizes the family of continuous partial differential equations that comprise together the physical procedures of Laplacian and diffusion [42].

Suppose an image is represented as a 2-D function \( p : \mathbb{R}^2 \rightarrow C \) where \( C \) is a color space representation. Assume \( C \) is a finite color space isomorphic to \( \mathbb{R}^m \) for some \( m \), and possesses a distance metric \( d_c \) [43]. If \( G_\sigma(x,y) \) is the normalized 2-D Gaussian kernel with standard deviation \( \sigma \),
\[ G_\sigma(x, y) = \frac{1}{2\pi\sigma} \exp\left( -\left( \frac{x^2 + y^2}{2\sigma^2} \right) \right) \] …(3.1)

after that a convolution can be used to calculate the uniform Gaussian blurring of \( p \) with \( \sigma = \sqrt{2t} \) as follows

\[ p(x, y, t) = p(x, y) * G_{\sqrt{2t}}(x, y) \] …(3.2)

\[ \int \int p(\xi, \nu, 0) G_{\sqrt{2t}}(x - \xi, y - \nu) d\xi dv \] …(3.3)

where a scale parameter \( t \) extends \( p \). Bigger values of \( t \) correspond to coarser (i.e. more blurred), depiction of \( p \). Still, \( p(x, y, t) \) is also the solution to the heat conduction differential equation.

\[ \frac{\partial p}{\partial t} = \Delta p \] …(3.4)

where \( \Delta p \) is the spatial Laplacian of \( p \).

\[ \Delta p = \frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial y^2} \] …(3.5)

The original image \( p(x, y) \) is the initial condition \( p(x, y, 0) \) of the heat conduction equation and as \( t \) and so \( \sigma \) tends to 0 the normalized Gaussian in Equation (3.1) converges to the Dirac delta \( \delta(x, y) \) [43].
A method of diffusion is defined by the above differential equation. Where the original space-variant intensity of \( p \) is redistributed spatially over time, with the rate of change is maximum where the curvature as calculated by the Laplacian, is highest. The prior equation explains the cutback of such maxima as \( t \) increases because a sharp positive (negative) peak has a large negative (positive) Laplacian. \( p(x, y, t) \) converges to a constant value incapable (if its initial energy is finite and its region is bounded) as the scale / time parameter \( t \) goes to infinity. The advantageous properties of conservation are present in diffusion: the value of the Integral \[ \int p \, dx \, dy \] is preserved or conserved; and causality: new maxima or minima cannot be fixed; only detached [42, 43].

By preserving these beneficial properties, diffusion can be generalized by integrating a conductance function \( c \) bounded between 0 and 1 that can be a function of space, time, and the image itself [39]:

\[
\frac{\partial P}{\partial t} = \nabla \cdot (c(p) \nabla p) = c(p) \Delta p + \nabla c(p) \cdot \nabla p \tag{3.6}
\]

by \( c(p) = c_e(p) c_a \), where \( c_a \) indicates a priori coherence information (i.e., \( c_a \) is a function of \( x \) and \( y \) only) and \( c_e \) depends on a space- and time-variant assessment of the
coherence of $p$. Anisotropic diffusion has been most intensely analyzed where the dependence of $c$ on $p$ is a monotonically decreasing function of the magnitude of the gradient $|\nabla p|$.

Unlike isotropic diffusion, anisotropic diffusion does not change the obvious position of edges. As an alternative, it deteriorates edge contrast over time, with the weaker edges disappearing first. For the short period, it can sharpen the transition as it can enhance the magnitude of the gradient of the image around an edge. However ringing artifacts cannot be introduced as this sharpening does not breach causality [43].

3.2. Color Space

A color space is a process that can be used to specify, create and visualize color. Human beings recognize a color by its attributes of brightness, shade and colorfulness. The colors in the computer are explained using the quantities of red, green and blue phosphor emission, which are needed to match a color. On a printing paper, a printing press can generate a precise color in terms of the reflectance and absorbance of cyan, magenta, yellow and black inks. Thus a color is generally denoted by three co-ordinates, or parameters. These parameters specify the site of the color within the color space being used. They do not denote the color because it depends on what color space is being used.

A color model is a theoretical mathematical model which characterizes the means that colors can be represented as tuples of numbers, usually as three or four values or color components (e.g. RGB and CMYK are color models). Though a color model with no related mapping function to an absolute color space is a more or less arbitrary color system with has no relationship to any globally understood system of color interpretation. An explicit "footprint" within the reference color space is generated by adding a certain mapping function between the color model and a certain reference color space and that "footprint" is known as a gamut, and in
permutation with the color model, defines a new color space. For instance Adobe RGB and sRGB are two diverse absolute color spaces, both based on the RGB model.

### 3.2.1. YCᵦCᵡ Color Space

In video and digital photography system, YCᵦCᵡ or Y’CᵦCᵡ, occasionally written as YCᵦCᵦ or Y’CᵦCᵦ, which is a family of color spaces. As part of the International Telecommunication Union – Recommendations sector BT.601 (ITU-R BT.601), Y’CᵦCᵡ Color Space was developed for worldwide digital component video standard and it is used in television transmissions. Here Y’ is represented as luma component and Cᵦ and Cᵡ indicate the blue-difference and red-difference chroma components. Y’ (With prime) is categorized from Y which is luminance, that means the light intensity is non-linearly encoded with gamma correction.

Y’CᵦCᵡ is a method for encoding RGB information and it is not a complete color space. The actual color displayed depends on the actual RGB colorants, which is used to display the signal. Thus a value expressed as Y’CᵦCᵡ is only anticipated if standard RGB pigments or an ICC profile are used. YCᵦCᵡ and Y’CᵦCᵡ are the two practical estimations to color processing and perceptual uniformity, where the primary colors succeeding roughly to red, green and blue are processed into relatively important information. Through this, the consequent video/image processing, transmission and storage can do operations and confirm errors in a significant way. The luma signal (Y’) is divided using Y’CᵦCᵡ, and the separated signal can be gathered with high resolution or transmitted at high bandwidth. To enhance the system
efficiency, the band width of the two chroma components (C_b and C_r) can be reduced, sub-sampled, compressed or treated discretely.

Based on the recommendation used, the transformation formula for this color space varies. We use the recommendation Rec 601-1, which provides values for red, green and blue color. For red, the value is 0.2989. For green, the value is 0.5866. For blue, the value is 0.1145.

![Figure 3.1 YC_b C_r Color Space](image)

RGB to YC_b C_r transformation:

\[
\begin{align*}
Y &= 0.2989 \, R + 0.5866 \, G + 0.1145 \, B \\
C_b &= -0.1688 \, R - 0.3312 \, G + 0.5000 \, B \\
C_r &= 0.5000 \, R - 0.4184 \, G - 0.0816 \, B
\end{align*}
\]  

...(3.7)
3.2.2. NBS/ISCC color space

The Inter-Society Color Council and National Bureau of Standards (ISCC-NBS) system of Color Designation is a system used for naming colors and it is related on a set of 12 basic color terms along with a small set of adjective modifiers. This system was first authorized in 1930s. The first chairman of the ISCC in 1932 recommended that the objective of the system is to be “a means of designating colors in the United States Pharmacopoeia, in the National Formulary, and in general literature such designation to be sufficiently consistent as to be adequate, broad to be appreciated and commonplace to be understood and utilizable by science, art, and industry, at least in a common way, by the whole public.” The system presents a basis on which color definitions in the fashion and printing fields to Botany and Geology can be systematized.

NBS have developed the ISCC-NBS dictionary, following the recommendation of the Inter-Society Council. It includes 267 terms, which are achieved by associating five descriptors for lightness (very dark, dark, medium, light, very light), four for saturation (grayish, moderate, strong, vivid), three for brightness and saturation (brilliant, pale, deep), and twenty-eight for hues built from a basic set (red, orange, yellow, green, blue, violet, purple, pink, brown, olive, black, white, gray). Nevertheless, as mentioned in, certain dictionaries frequently undergo many drawbacks like the lack of both a well-defined color vocabulary and a correct transformation to a different color space. Frequently colors are treated under three headings hue, lightness, and saturation.
3.3. Filtering

Image filtering is a method which is used to improve the image or else it can modify, warp, and damage images. By using this method, we can remove the noise, sharpen contrast, and highlight the contours of the images.

3.3.1. Wiener Filtering

It is a noise filter which relies on Fourier iteration. Short computational time is one of the main advantages to find out a solution.

Suppose there is some uncorrupted signal \( u(t) \) which is to be measured. The output signal is corrupted due to imperfection in equipments and this is the error that arises while measuring. The signals can be corrupted in two ways: In the first way, the equipment can convolve, or 'smear' the signal. This happens due to the improper equipment, delta function response to the signal. Let \( s(t) \) be the smear signal and \( r(t) \) be the known response that cause the convolution.

\[
s(t) = \int r(t-\tau)u(\tau)d\tau \quad \text{or} \quad S(f) = R(f)U(f)
\]  

\( ...(3.8) \)

where \( S, R, U \) are Fourier Transform of \( s, r \) and \( u \).

The second reason of signal corruption is due to unknown background noise. The measured signal \( c(t) \) is a sum of \( s(t) \) and \( n(t) \):

\[
c(t) = s(t) + n(t)
\]  

\( ...(3.9) \)

By dividing \( S(f) \) by \( R(f) \), we can deconvolve \( s \) to find \( u \), i.e. \( U(f) = \frac{S(f)}{R(f)} \) in the absence of noise \( n \). To deconvolve \( c \) where \( n \) is present then one need to find an optimum filter function \( \phi(t) \) or \( \Phi(f) \), then the noise is filter out by \( \Phi(f) \) and provides a signal \( \widetilde{u} \) by \([44]:\)
\[ U(f) = \frac{C(f)\Phi(f)}{R(f)} \quad \text{...(3.10)} \]

where \( \bar{u} \) is as close as possible to the original signal. If their differences square is close to zero then \( \bar{u} \) and \( u \) are similar, i.e. \( \int_{-\infty}^{\infty} |\bar{u}(t) - u(t)|^2 dt \) or \( \int_{-\infty}^{\infty} |U(f) - U(f)|^2 df \) is minimized.

Substituting equation (3.8), (3.9) and (3.10), the Fourier version becomes:

\[ \int_{-\infty}^{\infty} |R(f)|^{-2} |S(f)|^2 |1 - \Phi(f)|^2 + |N(f)|^2 |\Phi(f)|^2 df \quad \text{...(3.11)} \]

after rearranging. If the above integral is minimum for every value of \( f \) then it is a best filter. This is when,

\[ \Phi(f) = \frac{|S(f)|^2}{|S(f)|^2 + |N(f)|^2} \quad \text{...(3.12)} \]

Now, \( |S(f)|^2 + |N(f)|^2 \approx |C(f)|^2 \) where \( |C(f)|^2 \), \( |S(f)|^2 \) and \( |N(f)|^2 \) are the power spectrum of \( C, S \) and \( N \) respectively. Therefore,

\[ \Phi(f) \approx \frac{|S(f)|^2}{|C(f)|^2} \quad \text{...(3.13)} \]

Figure 3.2 depicts a graph of \( |C(f)|^2 \). It can be seen that \( |S(f)|^2 \) and \( |N(f)|^2 \) can be used to estimate the fact from the graph.
Based on the above theory, using Fourier Transform, it is clearly noted that we can write a program to Wiener Filter signal from noise. Mean-Squared Method is another method to Wiener filtering the signal without Fourier transforming the data [44].

**Mean-Squared Method:** According to the Mean-squared Methods, Wiener Filter is one that relies on the least-squared principle, i.e. the filter decreases the error between the actual and the desired outputs. Narrowing the width of the distribution curve eliminates noise as in figure 3.2.

For doing this filter, at first we have to find variance of the data matrix and then around the matrix a box of certain size is sent through pixel by pixel. The local mean and variance of each box is found. From the subsequent formula, the filtered value of each pixel can be evaluated [44]:

![Figure 3.2 Plot of power spectrum of signal plot noise](image-url)
\[ A_{ij} = \mu_{ij} + \frac{\sigma_{ij}^2 - s^2}{\sigma_{ij}} (N_{ij} - \mu_{ij}) \]  \hspace{1cm} (3.14)

where \( A_{ij} \) is the filtered signal, \( \mu_{ij} \) is the local mean, \( \sigma_{ij} \) is the local variance, \( s^2 \) is the noise variance of the entire data matrix, and \( N_{ij} \) is the original signal value.

From the above formula, it can be vividly viewed that, if the original signal and local variance are the same then the local mean will be treated as the filtered value, otherwise the local mean will be filtered to provide a higher/lower intensity signal basing on the difference. In addition to that, if the local variance and matrix variance are equal, which is around 1 (i.e. only noise exists in the box) then the filtered signal will be that of the local mean, which must be near to zero. When the local variance is greater than the matrix variance the signal will be however amplified \([44]\). The data is filtered as the box passes the whole matrix, determining the result to each pixel with the help of the above formula.

### 3.3.2. Gaussian Filtering

Gaussian filter is a linear filter used for smoothing the images. It combines both low pass filtering and differentiation to detect edges or computing orientation features in digital images. Marr and Hildreth promoted the necessity for an operator that can be tuned to detect edges at a particular scale. Their technique depends on filtering the image with a Gaussian kernel chosen for a particular edge scale. The Gaussian smoothing operation aids to band-limit the image to a small range of frequencies, and reducing the noise sensitivity problem when detecting zero crossings. Over a variety of scales, the images are filtered and the Laplacian zero crossings are estimated at each. This generates a set of edge maps as a function of edge scale. Each edge point can be measured to survive in a region of scale space, where the edge point location is a function of \( x, y \) and \( s \). For refining and analyzing edge maps, scale space has been effectively used.
Certainly Gaussian has some advantageous properties which facilitate this edge detection procedure. First the Gaussian function available in both the spatial and frequency domains are smooth and localized, also it provides an excellent compromise between the need for avoiding false edges and for reducing errors in edge position. Indeed the Gaussian is the only real-valued function, so that the product of spatial and frequency-domain spreads are reduced. Because of its differential and smoothing behavior, the Laplacian of Gaussian fundamentally acts as a band pass filter. Second property implies that the Gaussian is separable, which helps the computation very efficient.

Without the scaling factor, the Gaussian filter can be expressed as

$$g_c(x, y) = \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right)$$

...(3.15)

Its frequency response $G(\Omega_x, \Omega_y)$ is also Gaussian:

$$G(\Omega_x, \Omega_y) = 2\pi\sigma^2 \exp\left(-\frac{\sigma^2}{2}(\Omega_x^2 + \Omega_y^2)\right)$$

...(3.16)

The parameter $G$ is inversely related to the cutoff frequency. Their computation order can be interchanged, since the convolution and Laplacian operations are both linear and shift invariant.

$$\nabla^2\left[f_c(x, y)^* \cdot g_c(x, y)\right] = \left[\nabla^2 g_c(c, y)^* \cdot f_c(x, y)\right]$$

...(3.17)

Here we are taking the linear operator as the derivative hence Gaussian filtering following by differentiation is the same as filtering with the derivative of a Gaussian. From the right-hand side of equation (3.17), there usually more proficient computation is provided since
\( \nabla^2 g_c(x, y) \) can be organized in advance as a result of its image independence. Hence the Laplacian of Gaussian (LoG) filter \( h_c(x, y) \) has the subsequent impulse response.

\[
h_c(c, y) = \nabla^2 g_c(x, y) = \frac{x^2 + y^2 - 2\sigma^2}{\sigma^4} \exp \left( -\frac{x^2 + y^2}{2\sigma^2} \right)
\]

... (3.18)

### 3.4. Color Quantization

Color quantization is the process that reduces the number of distinct colors in an image, so that the new image looks like the original image. In 1970s computer algorithms are used, for performing color quantization on bitmaps have been studied. Color quantization is momentous, because, instead of displaying images with many colors on devices, the color quantization can display only a limited number of colors, generally due to memory limitations, and that enables proficient compression of certain types of images. Recently it is extensively used, and in many multimedia applications, it reduces the workload of massive image data on storage and transmission bandwidth [45]. A color-quantized image can be considered as a degraded version of the original full-color image. Color quantization contains two important points:

- For generating a color map (or palette), a small set of colors (typically 8-256) is selected from the (224) promising combinations of red, green and blue (RGB).
- Each color picture element in the color image is mapped to any one of the colors in the color map.

Furthermore as formerly mentioned, this chart (mapping) should decrease the variation between the quantized image and original image [45].

### 3.5. Clustering
Separating a group of information points into a number of sub-groups is called clustering analysis, where the objects within a bunch are treated as subgroup. This subgroup shows some degree of resemblance. Clustering analysis has been made a significant role in solving many problems in pattern recognition and image processing. Clustering methods can be measured as either hard (crisp) or fuzzy based on whether a pattern information belongs entirely to a single cluster or to several clusters with diverse degrees. A membership value of zero or one is assigned to each pattern data (feature vector) in hard clustering, but in fuzzy clustering, a value between zero and one is assigned to each pattern by a membership function. Usually for hard counterparts, the fuzzy clustering methods can be considered to be superior, as they can signify the relationship between the input pattern data and clusters more naturally.

3.5.1. Fuzzy K Means Clustering

Clustering a data $X \subseteq R^N$ indicates that the data is divided into $K$ clusters such that every cluster is packed in and far away from adjacent clusters. The goal is attained by reducing the distances between the cluster center and patterns that cluster. By this principle, the hard $K$-means algorithm minimizes the ensuing objective function [46]:

$$J = \sum_{k=1}^{K} \sum_{x_i \in F_k} d(m_k, x_i)$$  \hspace{1cm} \ldots(3.19)

From the above equation, $d(m_k, x_i)$ denotes the distance measure between the center $m_k$ of the cluster $F_k$ and the pattern $x_i \in X$. Eqn. (3.19) can be rewritten as

$$J = \sum_{k=1}^{K} \sum_{i=1}^{n} \mu_k(x_i) d(m_k, x_i)$$  \hspace{1cm} \ldots(3.20)
where \( \mu_k(x_i) \in \{0,1\} \) is the characteristic function, i.e., \( \mu_k(x_i) = 0 \) if \( x_i \not\in F_k \), else \( \mu_k(x_i) = 1 \). Each pattern may belong to more than one cluster, when the clusters are overlapped, i.e., \( \mu_k(x_i) \in [0,1] \) Therefore, \( \mu_k(x_i) \) should be interpreted as a membership function rather than the characteristic function. So the objective function (3.20) can be altered as:

\[
J = \sum_{k=1}^{K} \sum_{i=1}^{n} \mu_k^q(x_i) d(m_k, x_i) \quad \text{...(3.21)}
\]

where \( \mu_k(x_i) \in [0,1] \) is a fuzzy membership function, and \( q \) is a constant known as the index of fuzziness that controls the amount of fuzziness. As the minimization of the objective function (3.21) may lead to a trivial solution, the subsequent two constraints are satisfied while reducing the objective function:

\[
\sum_{i=1}^{n} \mu_k(x_i) > 0 \quad \forall k \in \{1,2,...,K\} \quad \text{...(3.22)}
\]

\[
\sum_{k=1}^{n} \mu_k(x_i) = 1 \quad \forall i \in \{1,2,...,n\} \quad \text{...(3.23)}
\]

The first restriction assures that there is no vacant cluster, and the second restriction deals with the circumstances that each pattern wants to share its membership with all the clusters such that the sum of memberships is equal to one [47]. By differentiating the objective function (3.21) through the limitations (3.22) and (3.23), we obtain

\[
\mu_k(x_i) = \frac{1}{\sum_{h=1}^{K} \left( \frac{d(m_k, x_i)}{d(m_h, x_i)} \right)^{2/(q-1)}} \quad \forall i \in \{1,...,n\}, k \in \{1,...,K\} \quad \text{...(3.24)}
\]

\[
m_k = \frac{\sum_{i=1}^{n} \mu_k^q(x_i) x_i}{\sum_{i=1}^{n} \mu_k^q(x_i)} \quad k = 1,2,...,K \quad \text{...(3.25)}
\]
To modify the memberships and the cluster centers, Eqn. (3.24) and (3.25) are used in an iterative manner. The updating goes on until the changes in the membership values of all the patterns become insignificant or the required number of iterations is over (Figure 3.3).

The worst-case time complexity of the algorithm is as follows: We need $O(nN)$ computations for finding the gap between the cluster center and all the patterns. The number of computations $O(nNK)$ is needed for all the clusters. If the clustering needs $T$ iterations, then the worst situation complexity is $O(nNKT)$.

Input:

1. X is a set of input data.
2. q is fuzziness index where $q \in (1, \infty)$
3. K denotes number of clusters.
4. A distance measure $d(m_k, x_i) = d(m_k - x_i)' A^{-1} (m_k - x_i)$
   where $A$ is a positive definite matrix.
5. $\varepsilon$ is a small positive constant and approximate matrix norm
6. T represents maximum number of iterations
7. U is an $n \times K$ matrix, the $i^{th}$ row and the $k^{th}$ column indicates $\mu_k(x_i)$.

The k-means algorithm steps:

1. Assign $t=0$
   Set $t = t + 1$
   Randomly initiate the fuzzy k-partition of $u^t$

2. Calculate the cluster center $m_k$ using

$$m_k = \frac{\sum_{i=1}^{n} \mu_k^q(x_i) x_i}{\sum_{i=1}^{n} \mu_k^q(x_i)}$$
3. Update $u^{t+1}$ by calculating $u^t$ as

$$I_k = \{k \mid 1 \leq k \leq K; d(m_k, x_i)\} = 0$$

If $I_k = 0$

$$\mu_k(x_i) = \frac{1}{\sum_{h=1}^k \left[ \frac{d(m_k, x_i)}{d(m_h, x_i)} \right]}$$

Else $\mu_k(x_i) = 0 \quad \forall \ k \in \{1, 2, \ldots, k\} - I_k$

and $\sum_{k \in I_k} \mu_k(y_j) = 1$

4. If $|u_t - u_{t-1}| > \varepsilon$ or $t < T$ then stop.

Otherwise return to step 2.

3.6. Jet Area

With the help of viewing the regurgitant jet area in the receiving chamber we can get a rapid screening of the presence and direction of the regurgitant jet and a semi-quantitative evaluation of its severity. Different factors of physiology, anatomy and technique exercise an authority on the size of the regurgitant area and modifying its precision as an index of regurgitation severity [48]. The size of the jet can be influenced by the instrument factors, exactly pulse repetition frequency (PRF) and color gain. Removing the random color speckle from non-moving regions is utilized in the standard method by Nyquist limit (aliasing velocity) of 50-60 cm/sec and a color gain.

Usually the more MR is implied by the large jets that pull out deep into the left atrium. The small thin jets that emerge just ahead of the mitral leaflets do not pull out so much deep. On account of sundry technical and hemodynamic restrictions which are observed previously [48] the correlation is poor between jet area and MR severity. The patients who have low blood
pressure (BP) can detect small eccentric color jet area and LA pressure owing to chronic severe MR while the jet area for high blood pressure patients with mild MR is enormous. On account of the above reasons the estimation of severity of MR by “eyeballing” or planimetry of the MR color flow jet area is not recommended. Though small, non-eccentric jets covering an area less than 4.0 cm² or less than 20% of LA area trace mild MR frequently. On the opposing huge jets that pierce into the pulmonary veins are probably hemodynamically vital.

3.7. Vena Contracta

Vena Contracta (VC) is the narrowest part of a jet that appears at or just down the orifice. High velocity and laminar flow are the main characteristics of the vena contracta. To some extent the Vena Contracta is smaller than the orifice of anatomic regurgitant orifice. The Vena Contracta cross sectional area represents a measure of EROA and it is the narrowest area of actual flow. The size of the Vena Contracta is not influenced by flow rate and driving pressure for fixed orifice. When its regurgitant orifice is dynamic the vena contracta is likely to change with hemodynamics or during the cardiac cycles [49]. The regurgitant orifice in MR might not be circular and is extended along the mitral coaptation line frequently. Numerous studies have exposed that the width of the Vena Contracta is exact in estimating the severity of MR, either by Transthoracic or Transesophageal Echocardiography [50 - 52]. For estimating the severity of MR both TTE and TEE aid in determining exactly the width of the Vena Contracta which is in turn advantageous [50 - 52]. The parasternal views support in the consistency of the width of the Vena Contracta in long-axis views and its cross-sectional area in short-axis views [52]. For mild MR, the cut off range will be Vena Contracta < 0.3 cm whereas 0.6 to 0.8 cm is the cut off range for severe MR [51, 52]. Moreover the intermediate values are apt to correlate well with moderate MR; there arises a need for the consumption of another technique for confirmation.
3.8. **Proximal Isovelocity Surface Area Method**

Proximal Isovelocity Surface Area method relates to the continuity equation. When a flow passes through a narrow orifice, there is flow convergence and flow acceleration, as it approaches the narrowest region. PISA is defined as the hemisphere surface area at the aliasing region of the flow convergence. PISA enhances as the flow increases and also with lower aliasing velocity. Smaller aliasing velocity has to be set, for diminishing the errors in measurement, to obtain higher PISA measurement with lower chance for errors. For the assessment of Valvular Insufficiency the PISA of a regurgitant color flow jet is used [53, 27, 54]. PISA depends on the hemodynamic principles where the blood flows through a small circular orifice in a flat plate. The flow acceleration is proximal to the orifice.

The basic idea behind the proximal convergence technique is the conservation of mass, with the assumption that, in the region proximal to a regurgitant orifice, flow is laminar and accelerates smoothly, forming concentric shells of rising the velocity and reducing the surface area (Hydrodynamic theory calculates that the flow converges in the direction of a restrictive orifice as a sequence of isovelocity shells with decreasing surface area and rising velocity) [55]. This region of higher velocity and lesser flow dimension is known as Vena Contracta [21]. Theoretically the flow convergence region proximal to a discrete regurgitant orifice in a flat planar surface is a hemispheric volume in which flow speeds up towards the regurgitant orifice beside the radial stream lines. This zone of proximal flow acceleration is made up of concentric hemispheric shells of identical and accelerating velocities (velocity isopleths).
Figure 3.3 Illustration of Proximal Flow Convergence and the regurgitant jet

The hemisphere surface area is calculated from the fundamentals as two pi radius squared. Color flow Doppler can be used to calculate hemisphere area. The volume of regurgitant blood flow (cm$^3$/s) is provided by multiplying the hemisphere area (cm$^2$) with aliasing velocity (cm/s). To estimate regurgitant flow rate, the Proximal Flow Convergence method needs measurement of the distance from the level of the leaflet orifice to the position at which a given velocity arises [53], [56]. Flow rate ($Q$) can be measured at any of the converging shells as the product of the isovelocity surface area times the velocity at that shell are done according to the principle of conservation of mass orifice.

3.9. MATLAB

MATLAB is a software package intended for (among other things) data processing. MATLAB effortlessly handles matrix and complex arithmetic and it is both a powerful computational environment and a programming language. It is an enormous software package with several advanced built-in features, and it has emerged as a standard tool for several working in science or engineering disciplines.
It includes huge amount of numerical algorithms, an intuitive script language and excellent data-visualization abilities. MATLAB uses a compact but simple notation and it is very easy to add functions to it, so it is a perfect tool for quick prototyping. MATLAB depends on matrix and vector algebra; still scalars are indicated as 1x1 matrices. Thus vector and matrix operations are as simple as ordinary calculator operations. For technical computing, MATLAB is a high-performance language [2]. It combines computation, visualization, and programming in a convenient environment. It comprises math and computation, algorithm development, data acquisition, modeling, simulation, and prototyping, data analysis, exploration, and visualization, scientific and engineering graphics, application development, including building graphical user interfaces.

Benefits of MATLAB are an interpreted language for numerical computation. It permits one to carry out numerical calculations, and without the help of complicated and time consuming programming, the result can be visualized. Although a disadvantage of MATLAB is an interpreted language, it can be slow, and poor programming practices can make computation unacceptably slow. For preliminary and advanced courses in mathematics, engineering, and science, MATLAB is used as a computational tool in universities, industries and for research and development.

MATLAB is comprised of a group of applications and specific problem solutions called toolboxes. The set of MATLAB functions (M-files) is called Image Processing Toolbox that solves digital image processing problems. Sometimes other toolboxes such as signal processing, Neural Networks etc. are also used to harmonize the Image Processing Toolbox.

### 3.9.1. Image Processing Toolbox
The Image Processing Toolbox (IPT) is a set of functions that pulls out the capability of the MATLAB® numeric computing environment. For image manipulation, analysis, digital imaging, computer vision, and digital image processing, IPT provides an inclusive set of functions. The IPT capabilities comprise image file I/O, Spatial image transformations, Color space transformations, Linear filtering, Mathematical morphology, Texture analysis, Neighborhood and block operations, Pattern recognition, Image statistics, Image analysis and enhancement, Image registration, Deblurring, Region of interest operations and more.

An array plays a vital role in data structure, which is a component of MATLAB. Most of the images in MATLAB are stored as two-dimensional arrays (i.e., matrices), and when the image is displayed, each element in the matrix represents a pixel. For example if an image consists of 100 rows and 200 columns of different colored pixels that would be stored in MATLAB as a 100-by-200 matrix. The RGB color images need a three dimensional array, where the first plane represents the red pixel intensities, the second plane represents the green pixel intensities and the third plane represents the blue pixel intensities [3]. This principle provides the working of images in MATLAB analogous to functioning by any other model of matrix information and the complete power of MATLAB is accessible for image processing application. The four essential types of images in Image Processing Toolbox are Index images, Intensity images, Binary images, and RGB images.